

УДК XXX.XXX

Гамосов С. С.

Северо-Осетинский государственный университет  
имени Коста Левановича Хетагурова,  
362025, Россия, г. Владикавказ, ул. Церетели, 16,  
gamosov.gs@yandex.ru

## Концепция Seq2Seq для нейронного перевода

**Ключевые слова:** Нейронный Машинный Перевод, Gated Recurrent Unit, Модель Seq2Seq, Attention, Осетинский язык.

### Введение

За последние годы качество моделей машинного перевода резко выросло, связано это с началом использования в них рекуррентных нейронных сетей (RNN). Они позволили снизить затраты на выявления лингвистических закономерностей языков и дорогостоящую разработку алгоритмов для их обработки, что использовалось в RBMT (Rule-Based Machine Translation). В свою очередь NMT (Neural Machine Translation) довольно хорошо справляется со своей задачей, но все таки ему до сих пор не удается полностью сместить SMT (Statistical Machine Translation). Хотя количество статистического перевода в современных переводчиках снизилось почти до минимума. Теперь уже при построении хорошей модели машинного перевода большую часть работы стоит сконцентрировать, на особенности языка, его структуре, связи между словами и их значениями. Решением данного перечня проблем и являются модели sequence-to-sequence ([4], [5], [6]).

Seq2Seq - это семейство подходов машинного обучения, используемых для обработки языка. Первоначальный подход Seq2Seq, который в процессе породил целое семейство методов, был разработан Google для использования в машинном переводе [11]. Как уже можно заметить за последнюю пару лет коммерческие системы стали удивительно хороши в переводе - посмотрите, например, Google Translate, Яндекс-Переводчик, переводчик DeepL, переводчик Bing Microsoft. Данный подход хорош не только в нейронном переводе. Текстовое описание к изображению и обобщение (summarize) текста - вот не полный список задач, в которых используются данная модель.

В статье рассматривается задача нейронного машинного перевода - NTM (Neural Machine Translation). Формальная цель поставленной проблемы - это перевод предложений с одного естественного языка на другой с помощью машины. Так же помимо теоретического описания, построения, были реализованы модели и проведен поверхностный анализ качества полученных результатов.

# Описание модели NTM

Нейронный машинный перевод (NMT, Neural Machine Translation) - это радикальное изменение подходов к машинному переводу. С одной стороны, NMT использует непрерывные представления вместо дискретных символьных представлений. С другой стороны, NMT использует единую большую нейронную сеть для моделирования всего процесса перевода, избавляя от необходимости чрезмерного проектирования функций. Помимо своей простоты, NMT добился высочайшей производительности на различных языковых парах. На практике же NMT также становится ключевой технологией многих коммерческих систем, как уже было сказано выше.

В качестве подхода к машинному переводу, основанного на данных, NMT использует вероятностную структуру. С математической точки зрения, цель NMT состоит в том, чтобы оценить неизвестное условное распределение  $P(y|x)$  с учетом набора данных  $\mathcal{D}$ , где  $x$  и  $y$  - случайные величины, представляющие исходный ввод и целевой вывод соответственно.

Переводить можно последовательность на разных уровнях. В качестве единицы перевода можно взять предложения, абзацы или тексты. В статье основной единицей будет являться предложение. Благодаря такому уточнению, модель NMT можно рассматривать как модель sequence-to-sequence.

На вход подается предложение  $X = \{x_1, x_2, \dots, x_T\}$  и целевое предложение  $Y = \{y_1, y_2, \dots, y_T\}$ . Благодаря таким обозначениям, можно рассматривать перевод как нахождение целевой последовательности, которая является наиболее вероятной с учетом входных данных. Если более формально, то задача состоит в том, чтобы найти целевую последовательность, которая максимизирует условную вероятность  $P(y|x)$ . Однако этого не достаточно, нужен еще некоторый параметр  $\theta$ , от которого и будет изменяться условная вероятность в процессе обучения.

$$y' = \arg \max_y P(y = Y | x = X; \theta)$$

Таким образом, чтобы построить модель машинного перевода, нам нужно ответить на три вопроса:

1. Моделирование - Как выглядит модель для  $P(y|x; \theta)$ ?
2. Вывод - Как найти «лучший»  $y$ ?
3. Обучение - Как найти параметры  $\theta$ ?

Почти все модели нейронного машинного перевода используют стандартную парадигму Encoder-Decoder. Такая структура состоит из четырех основных компонентов: слоя Embedding, сетей Encoder и Decoder и уровня Classification.

Encoder - считывает исходную последовательность и создает ее представление. Decoder - использует исходное представление из Encoder для генерации целевой последовательности.

Для однозначности конца и начала предложения в последовательностях используются токены начала последовательности ( $\langle \text{sos} \rangle$  - start of sequence) и конца последовательности ( $\langle \text{eos} \rangle$  - end of sequence), чтобы меньше отцентрировать внимания на разборе особенностях языков.

Слой Embedding воплощает в себе сопоставление произвольной сущности (например, предложение или слово) некоторому вектору. Полученный дискретный результат обозначается как  $x_t \in \mathbb{R}^d$ , где  $d$  - размерность вектора. Затем embedding отправляются в другие слои сети.

Слой Encoder отображает исходный embedding в скрытое состояние  $h_t$ . Encoder должен уметь моделировать порядок и сложные зависимости, которые существовали в исходном языке. Рекуррентные нейронные сети (RNN) являются подходящим выбором для моделирования последовательностей переменной длины [6]. Опишем RNNs вычисления, выполняемые в слое Encoder, как:

$$h_t = \text{EncoderRNN}(x_t, h_{t-1})$$

В данном контексте под RNN может подразумеваться любая рекуррентная сеть. Например: LSTM [1] или GRU [2].

На каждом шаге итеративного процесса применяя полученную функцию EncoderRNN к входной последовательности, можно в конечном случае получить скрытое состояние  $h_S$  и использовать его в качестве представления для всей исходной последовательности (предложения). Затем передать его в Decoder.

Decoder в свою же очередь можно рассматривать как языковую модель, зависящую от  $h_S$ . Сеть Decoder извлекает необходимую информацию из выходных данных слоя Encoder, а также моделирует зависимости на больших расстояниях между целевыми словами. Однако в процессе работы с RNN было замечено, что архитектуры LSTM и GRU хоть и справлялись с моделированием зависимостей и особенностей языка, но все равно уступали статическому переводу. Поэтому решение внедрения механизма Attention является важной вехой в исследованиях архитектуры NMT [10]. Attention вычисляет релевантность каждого входного вектора значений на основе запросов и ключей (оригинальный перевод).

Учитывая начальный элемент последовательности  $y_0 = \langle \text{sos} \rangle$  и скрытое состояние  $s_0 = h_S$ , DecoderRNN сжимает полученное в результате декодирования в вектор состояния  $y_0, y_1, \dots, y_{t-1}, s_t \in \mathbb{R}^d$ :

$$s_t = \text{DecoderRNN}(y_{t-1}, s_{t-1})$$

Последний слой классификации предсказывает вероятностное распределение целевых элементов последовательности (возможный перевод слов в предложении от полученной модели). Классификация обычно представляет из себя линейный слой с функцией активации. Выше в представленных архитектурах используется функция активации softmax [2], [3]. Однако не стоит думать, что softmax является единственной функцией активации, которую

можно применять на данном этапе. В основном этот выбор зависит от поставленной задачи.

Предполагается, что словарный запас целевого языка равен  $V$ , а  $|V|$  - это количество слов в словаре. Размерность выхода  $z$  слоя Decoder, может быть абсолютно любой. Однако в конце концов, нужен вектор размерности  $|V|$ , чтобы получить вектор размерности  $|V|$  из вектора размера  $|z|$ , будем использовать линейный слой классификации. Затем на весь полученный результат применяем функцию активации softmax, чтобы гарантировать, что выходной вектор является допустимой вероятностью в диапазоне от 0 до 1.

После полного прохождения всех этапов, остается понять как сгенерировать перевод из полученной модели. В идеале хотелось бы найти целевую последовательность  $y'$ , которая максимизирует прогноз модели  $P(y|x = X; \theta)$  в качестве перевода, где  $X$  - входная последовательность. Однако из-за непомерно большого пространства поиска найти перевод с наибольшей вероятностью затратно. Поэтому NMT обычно использует локальные алгоритмы поиска, такие как Beam search [8], для поиска наилучшего перевода.

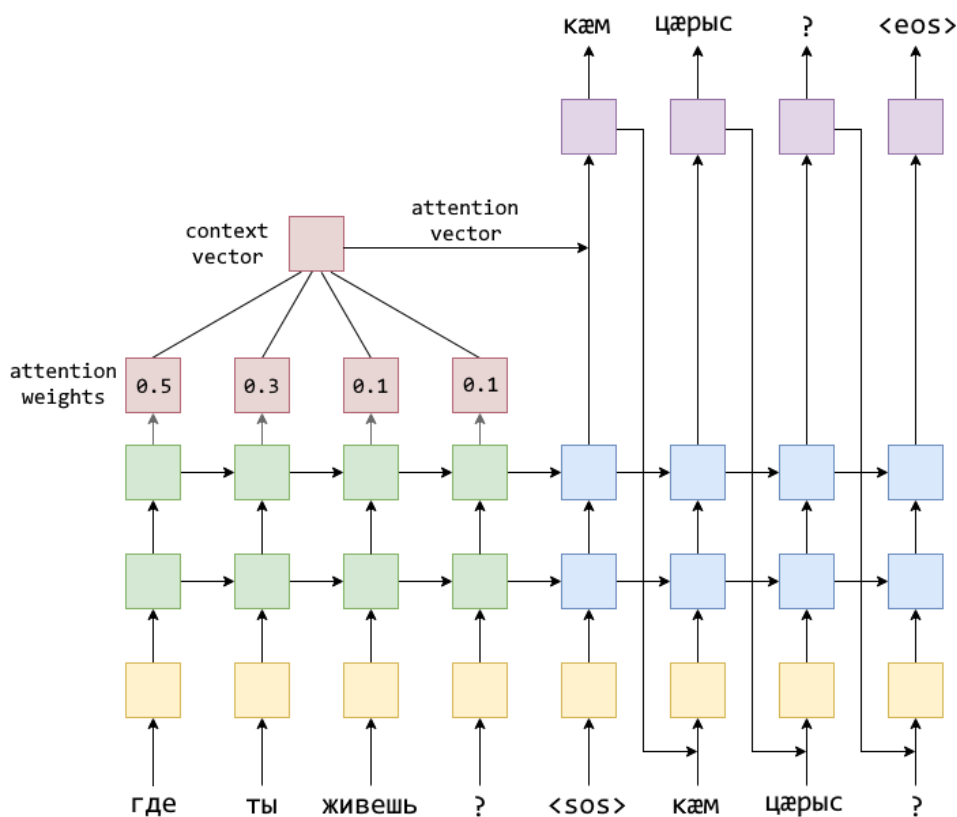


Рис. 1: Encoder-Decoder Seq2Seq модель, с механизмом Attention

Последним этапом построение модели машинного перевода является определение механизма обучения. Нейронные модели Seq2Seq обучаются предсказывать распределения вероятностей следующего элемента последовательности с учетом предыдущего контекста. На каждом шаге нужно максимизировать вероятность, которую модель присваивает правильному элементу.

Формально, давайте представим, что у нас есть обучающий экземпляр с входом  $X = \{x_0, \dots, x_T\}$  и целевой выход  $Y = \{y_0, \dots, y_T\}$ . Затем на временном шаге  $t$  модель предска-

зывает распределение вероятностей  $P^{(t)} = P(*|y_0, \dots, y_{t-1}, x_0, \dots, x_T)$ . Основная цель на этом этапе, чтобы модель присваивала вероятности 1 правильному элементу  $y_t$  и 0 остальным.

В таких случаях обычно использует максимальное логарифмическое правдоподобие (MLE) или категориальную кросс-энтропию (CCE) в качестве целевой функции обучения, которая является используемым методом оценки параметров распределения вероятностей. Формально, учитывая обучающий набор  $\mathcal{D} = \{<x(s), y(s)>\}_{s=1}^S$ , целью обучения является поиск набора параметров модели  $\theta$ , которые максимизируют логарифмическую вероятность на обучающем наборе:

$$\hat{\theta}_{\text{loss}} = \operatorname{argmax}(L(\theta)),$$

где логарифмическая вероятность определяется как:

$$L(\theta) = \sum_{s=1}^S \log(P(y^{(s)}|x^{(s)}; \theta))$$

Благодаря алгоритму обратного распространения ошибки можно эффективно вычислить градиент  $L$  относительно  $\theta$  (параметров модели). При обучении моделей NMT обычно используется алгоритм стохастического градиентного спуска (SGD). Вместо вычисления градиентов на полном обучающем наборе SGD вычисляет функцию потерь и градиенты на маленькой части обучающего набора. Простой оптимизатор SGD обновляет параметры модели NMT с помощью следующего правила:

$$\theta \leftarrow \theta - \alpha \nabla L(\theta)$$

где  $\alpha$  - скорость обучения. При правильно выбранной скорости обучения параметры NMT гарантированно сходятся к локальному оптимуму. На практике оказывается, что адаптивные оптимизаторы скорости обучения, такие как Adam [8], значительно сокращают время обучения, а точность почти не снижается.

## Эксперимент

### Датасет

Одним из главных аспектов для реализации хорошей модели перевода является качество и количество данных. Поэтому было принято решение проверять модели на двух датасетах. Большой датасет, содержащий пары русского и английского языка, даст возможность минимизировать ошибки в структуре построения. В свою очередь это увеличит шанс на удачную работу таких же моделей, но с другой парой языков. Второй датасет, содержащий пары русского и осетинского языка, гораздо меньше по размеру. Таким образом, можно сосредоточиться только на параметрах моделей и этапах их обучения, при использовании такой связки пар языков.

Тип языковой пары	Количество
Русский язык → Английский язык	100 000
Русский язык → Осетинский язык	1 878

Таблица 1: Количество пар в датасетах

Все материалы, который в последствие были использованы в качестве обучающих данных собраны на следующих ресурсах:

1. Русский язык → Английский язык |  $RUS \rightarrow ENG$ :

- (a) ManyThings.org - Двухязычные Пары предложений, разделенные табуляцией. Это выбранные пары предложений из проекта Tatoeba.

<http://www.manythings.org/anki/>

2. Русский язык → Осетинский язык |  $RUS \rightarrow OSS$  (Кусочно собран с разных ресурсов):

- (a) Проект *Tatoeba* - обширная база данных предложений и их переводов, постоянно пополняющаяся усилиями тысяч добровольных участников.

<https://tatoeba.org/ru/downloads>

- (b) Проект *Биоингвæтæ* - Билингвы подготовлены для чтения с помощью электронных словарей программы Lingvo.

<https://ironau.ru/bilingva/index.htm>

- (c) Ф.М. Таказов - Краткий русско-осетинский разговорник.

<https://ironau.ru/takazov/phrasebook2.htm>

- (d) Ф.М. Таказов - Самоучитель осетинского языка.

<https://ironau.ru/takazov/index.htm>

## Обработка данных

В процессе работы с данными были обнаружены некоторый перечень проблем.

Датасет  $RUS \rightarrow ENG$ , почти не имеет в себе проблем, кроме одной. В некоторых парах используется сокращения принятые в английском языке (Например: I am → I'm). Данная проблема не была решена. Поэтому в статье данная проблема игнорируется.

Так же в датасете  $RUS \rightarrow OSS$  были обнаружены символы «æ» (Код: 1237) и «æ» (Код: 230). Данная буква несет в себе одинаковую смысловую нагрузку в языке, следовательно просто заменим все символы на «æ» с кодом 230 (т.к. аналогичный символ используется на Осетинской Википедии).

Еще есть проблема с символом «тире». Два вида символов «—» и «-», тоже несут в себе одинаковую ценность, связи с этим заменяем их на какое-то одно в данном случае на второй символ из представленных («-»).

## Обучение

Были созданы две модели. Первая модель перевода с русского языка на английский, вторая с русского на осетинский. Если более конкретно говорить о конфигурации моделей, то в качестве разновидности архитектуры RNN в блоках использована GRU [4]. У моделей размер батча 64, размер Embedding слоя 256 токенов, количество эпох обучения является 15, функция потерь - Sparse Categorical Crossentropy.

Параметр	Переменная	Значение
<i>Размер словаря</i>	<code>max_vocab_size</code>	5000
<i>Размер Embedding слоя</i>	<code>embedding_dim</code>	256
<i>Размерность выходного слоя</i>	<code>units</code>	1024
<i>Количество эпох обучения выходного слоя</i>	<code>epochs</code>	15

Таблица 2: Используемые параметры моделей

Непосредственно перейдем к рассмотрению модели и их результатов.

Процедура обучения модели RUS-ENG заняла 1.5 часа на GPU NVidia RTX 2060 Super. На начальных этапах функция потерь имела значение около 7.462, но ошибка упала до значений ниже 2 после нескольких итераций. Последующие эпохи хоть и уменьшали значение, но не с такой скоростью, что была на начальном этапе. По окончании обучения итоговая ошибка составляет 0.2056.

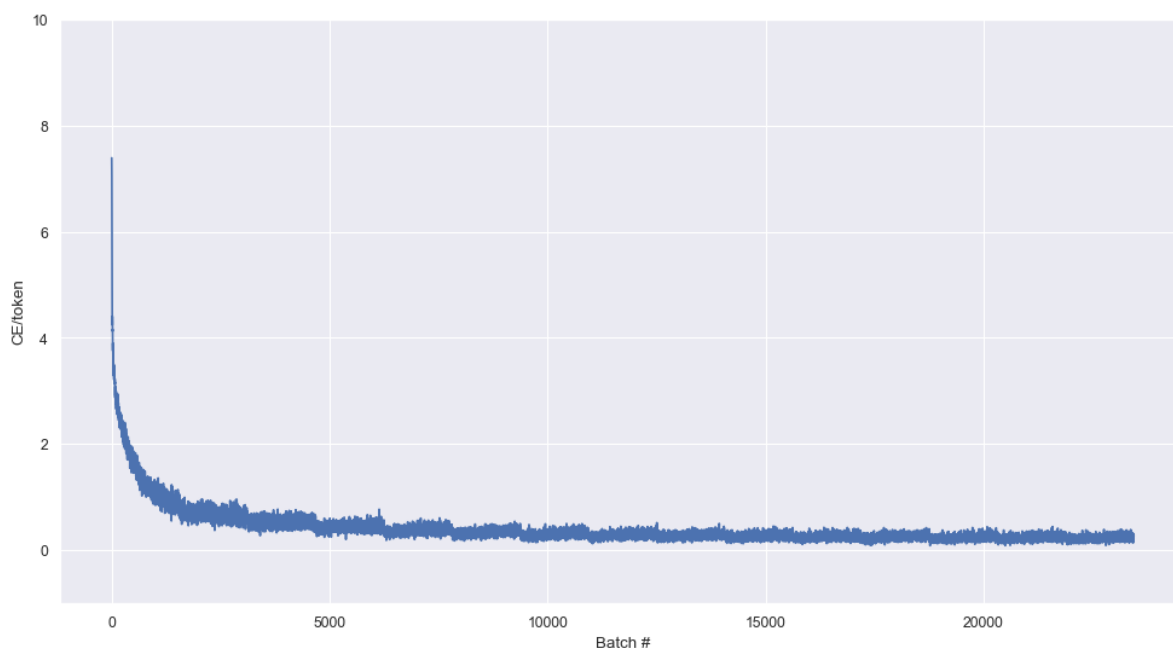


Рис. 2: Падающий график функции потерь во время обучения [RUS-ENG]

В датасете основанном на паре русского и осетинского языков около 1800 пар предложений. Это означает, что время на обучение модели значительно уменьшится. Процедура обучения RUS-OSS с таким же GPU составила 0.5 часа. Функция ошибка изначально находилась на отметке в 6.784 и после 15 эпох обучения упала до 0.217, что можно прекрасно увидеть на графике (Рис. 3).

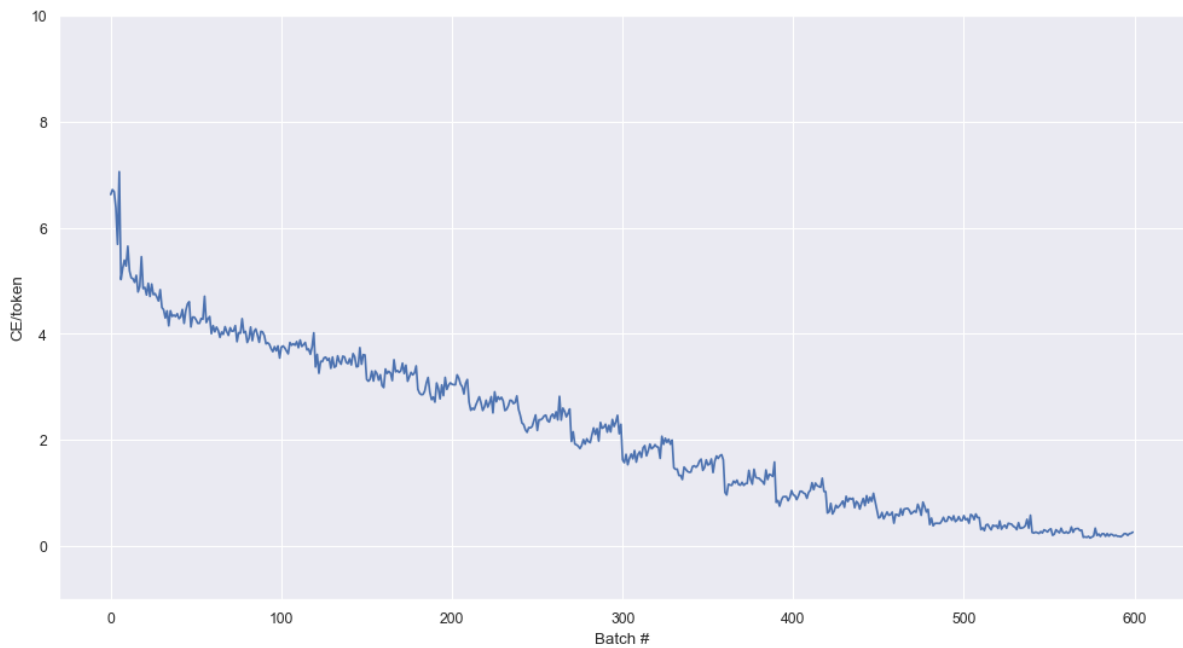


Рис. 3: Падающий график функции потерь во время обучения [RUS-OSS]

## Результат

Как оценить полученные модели? В качестве метрики качества перевода была выбрана BLEU [9]. Выходные данные BLEU всегда представляют собой число от 0 до 1. Это значение указывает, насколько текст-кандидат похож на справочные тексты, при этом значения, близкие к 1, представляют более похожие тексты. BLEU будет считать на основе n-грамм размера 4.

Посчитав средние значения BLEU-Score модели RUS-ENG, на текстовой выборки:

Тип	Значение
Среднее арифметическое	0.50529
Медиана	0.45761
Мода	1

Таблица 3: [Model RUS-ENG] Средние значения BLEU-Score

Полученные значение не особо дают ясности общей картине. Нельзя однозначно сказать, хорошая модель перевода у нас получилась или нет.

Тогда стоит рассмотреть сколько «хороших переводов» сгенерировала наша модель. Рассмотрим гистограмму по подсчету значений. В качестве абсциссы будет использоваться BLEU-Score одного предложения из тестовой выборки, ордината же в свою очередь представляет из себя количество. Высота полученных прямоугольников будет означать сколько всего было встречено предложений с одинаковым BLEU-Score. Для большей наглядности округлим показатели BLEU-Score до одного знака после запятой.

Смотря на полученный график (Рис. 4) видно, что довольно большая доля приходит на правильный перевод (BLUE-Score равный единице). В свою очередь весомая часть тестовой выборки дает, не однозначный показатель. По графику можно сказать, что почти больше



половины предложений переведено «наполовину правильно».

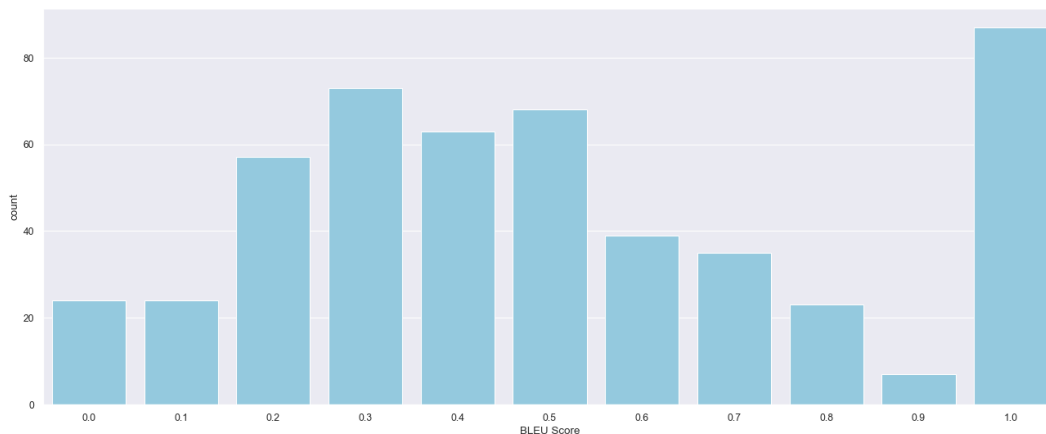


Рис. 4: [Model RUS-ENG] Гистограмма BLEU-Score для тестовой выборки

Само собой BLEU-Score довольно хорош в измерение качества перевода, однако стоит и рассмотреть конкретные примеры перевода.

Входное предложение «Открывай панель.». Модель выдает такой результат:

Original: Открывай панель.

Human Translation: open the panel .

Machine Translation: open the cage .

BLEU 4-gram: 0.5791460926441345

На месте, где должно быть слово «*panel*» находится «*cage*». Связано это с тем, что в словаре модели просто нет слова «*панель*» и соответственно его перевода тоже. В таких случаях модель может выдавать абсолютно случайны перевод.

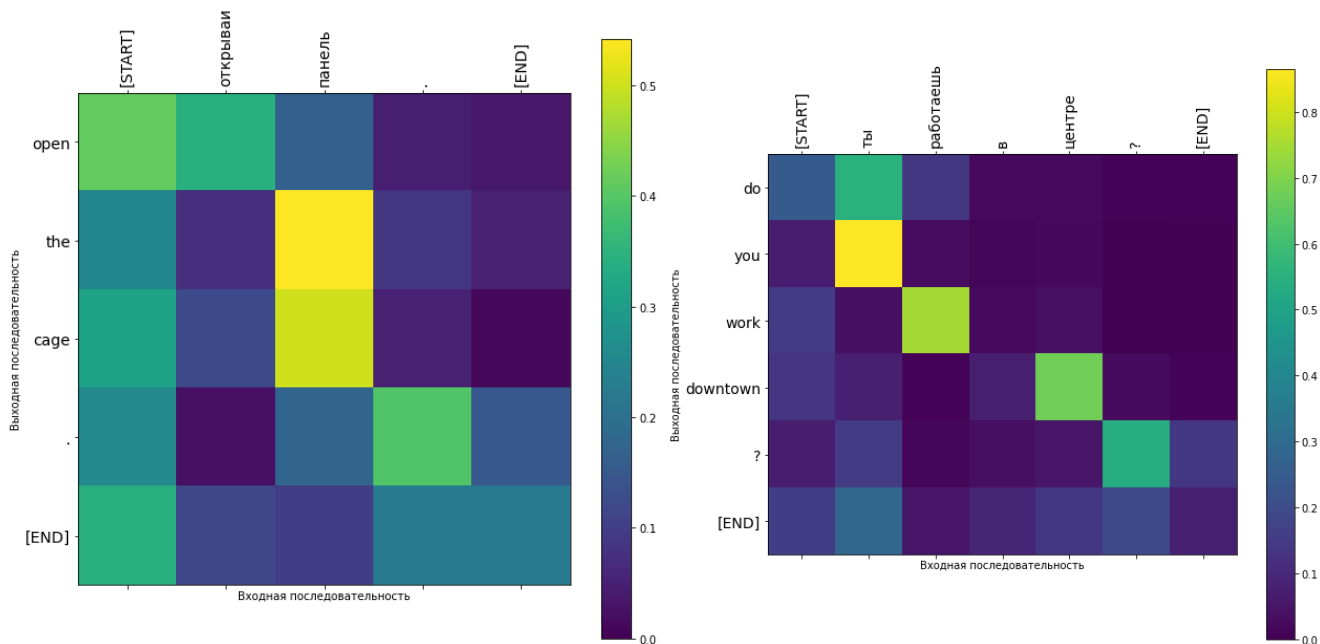


Рис. 5: (Левый) «Открывай панель.» (Правый) «Ты работаешь в центре?»

Исходя из этого можно сделать вывод, что структуру предложений, при наличие необходимого количества данных модель, распознает и генерирует сама. Однако если в словаре не

оказалось, слов, которые требуют перевода, модель ведет себя непредсказуемо.

Давайте рассмотрим более удачный перевод. Предложение «Ты работаешь в центре?».

Original: Ты работаешь в центре?

Human Translation: do you work downtown ?

Machine Translation: do you work downtown ?

BLEU 4-gram: 1.0

Модель довольно хорошо справилась с правилами построения вопроса на английском языке. Отсюда можно утверждать, что подбор «правильных» данных при обучении, сильно способствует улучшению качества модели.

На правой диаграмме (Рис.5), довольно хорошо видно, что хоть первое слово на вход это местоимение «*ты*», модель выдает большую вероятность на конструкции «*do you*», за счет обработки механизма Attention. Так же игнорирует предлог «*в*», и вовсе его не переводит.

Напрашивается вывод, что качество модели напрямую зависит от датасета, на котором она обучается. Сама модель довольно хорошо справляется с моделированием нюансов языка, на котором обучалась.

Давайте так же рассмотрим модель с парами русского и осетинского языков ( $RUS \rightarrow OSS$ ). Взглянув на гистограмму по подсчету значений BLUE-Score можно сразу заметить, что количество абсолютно неправильных переводов больше в разы по соотношению с неправильными, в отличии от предыдущей модели. Промежуточные показатели выглядят абсолютно не существенно по отношению к значениям BLUE равных нулю и единице.

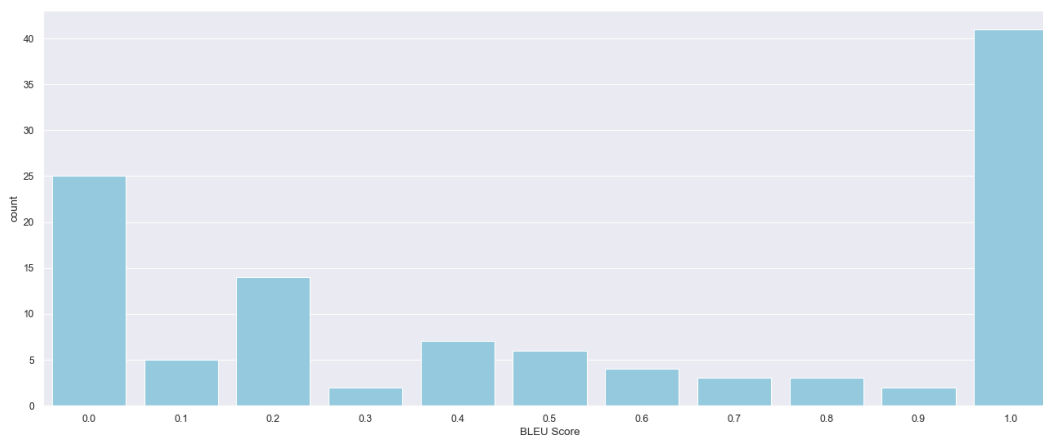


Рис. 6: [Model RUS-OSS] Гистограмма BLUE-Score для тестовой выборки

Посчитаем такие же средние значения для текущей модели:

Тип	Значение
Среднее арифметическое	0.5555
Медиана	0.6019
Мода	1

Таблица 4: [Model RUS-OSS] Средние значения BLUE-Score

Сразу видно, что средние показатели выше, по сравнению с моделью RUS-ENG, но качество перевода все равно ниже. Хотя и значения модели выглядят лучше, но по оценке BLEU

перевод все равно хуже. Как же так получается? На самом деле такое сравнение не совсем корректно, потому что количество пар предложений на которых модели обучались довольно сильно отличаются. Средние значения почти никакой конкретной информации нам не дают.

Поэтому всегда после общих оценок стоит рассмотреть конкретные переводы и сравнить их качество и постараться выделить некоторые особенности.

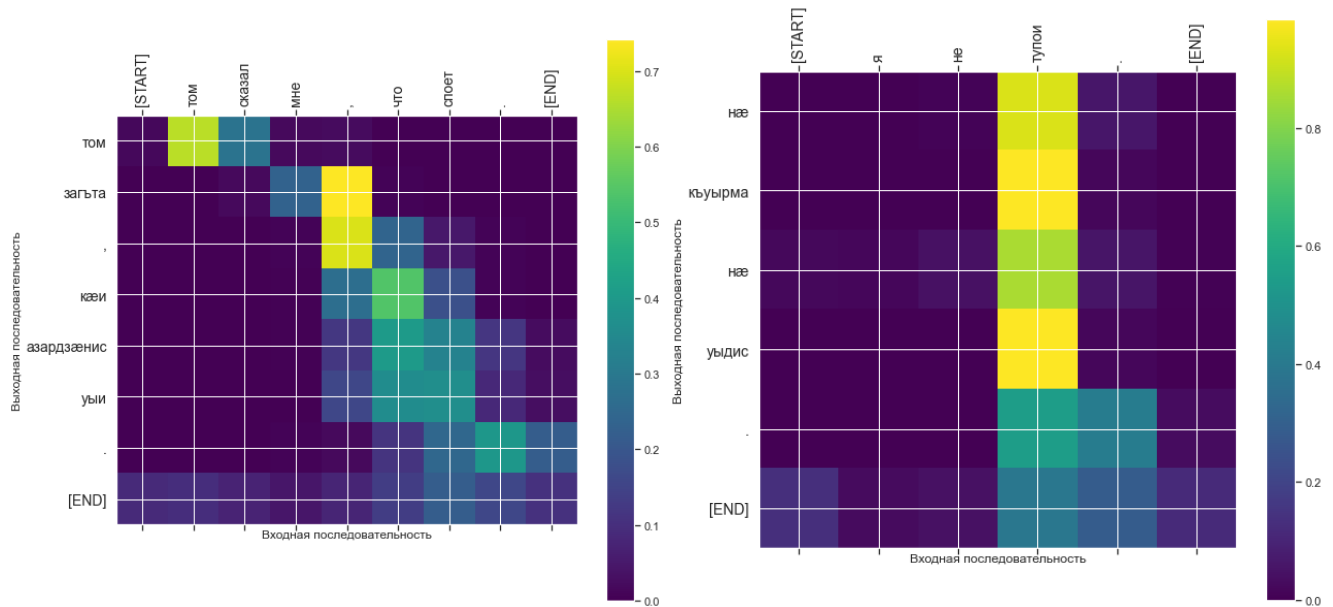


Рис. 7: Attention. (Левый) «Том сказал мне, что поёт.» (Правый) «Я не тупой.»

Original: Том сказал мне, что поёт.

Human Translation: том мын загъта, кæи азардæнис уыи.

Machine Translation: том загъта, кæи азардæнис уыи.

BLEU 4-gram: 0.8649141545451661

На левом Рис. 7 видно, что модель не особо придала значение местоимению «мне» и не перевело его. Оценка BLEU, хоть и не является идеальной, но смысл переводимого предложения остается верным. Так же видно, что структура построения осетинского предложения и русского гораздо ближе, чем между русским и английским, поэтому на графике можно заметить, что большие вероятности расположены на диагонали. Это говорит о том, что перевод происходит почти дословно.

Второй же случай показывает, что модель не справляется с предложениями как хотелось бы.

Original: Я не тупой.

Human Translation: æз къуырма нæ дæн.

Machine Translation: нæ къуырма нæ уыдис.

BLEU 4-gram: 0.5999350121649039

Хоть предложение и переведено по оценки BLEU почти на 50% верно, но смысл фразы потерялся полностью. На диаграмме (Правый Рис. 7), мы видим не диагональную подсветку вероятностей, а почти вертикальную прямую. Это говорит о том, что модель посчитала

только одно слово «тупой» как особо важное и почти все предложение переводило учитывая контекст только этого слова.

Основываясь на построенных моделях, можно сказать, что качество модели не только зависит от параметров самой модели, но и от данных на которых она обучалась. Модель обученная на данных RUS-ENG получилась гораздо лучше, чем модель на данных RUS-OSS.

## Вывод

Нейронный машинный перевод стал доминирующим подходом к машинному переводу как в исследованиях, так и на практике. В этой статье были рассмотрены широко используемые методы в NMT, включая моделирование, декодирование, интерпретацию, а также оценку данных.

Несмотря на большой успех, достигнутый NMT, все еще остается много проблем, которые необходимо изучить. Одна из таких это попытки интерпретировать NMT, такая проблема все еще актуальна. Понимание того, как и почему нейронный машинный перевод выдает такой результат, важно для определения недостатков и неточностей NMT.

Исходя из проведенных экспериментов с разными моделями машинного перевода, можно сделать вывод концепция Seq2Seq нейронного перевода, довольно хорошо справляется с построением правильного с точки зрения смысла и грамматики перевода, однако на прямую зависит от качество обучаемых данных.

# Список литературы

- [1] S. Hochreiter, J. Schmidhuber - Long Short-Term Memory; 1997.  
[https://www.researchgate.net/publication/13853244\\_Long\\_Short-term\\_Memory](https://www.researchgate.net/publication/13853244_Long_Short-term_Memory)
- [2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio - Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling; 2014.  
[https://www.researchgate.net/publication/269416998\\_Empirical\\_Evaluation\\_of\\_Gated\\_Recurrent\\_Neural\\_Networks\\_on\\_Sequence\\_Modeling](https://www.researchgate.net/publication/269416998_Empirical_Evaluation_of_Gated_Recurrent_Neural_Networks_on_Sequence_Modeling)
- [3] Ilya Sutskever - Training recurrent neural networks; 2013.  
[https://www.cs.utoronto.ca/~ilya/pubs/ilya\\_sutskever\\_phd\\_thesis.pdf](https://www.cs.utoronto.ca/~ilya/pubs/ilya_sutskever_phd_thesis.pdf)
- [4] Ri Wang, Maysun Panju, Mahmood Reza Gohari - Classification-based RNN machine translation using GRUs; 2017  
[https://www.researchgate.net/publication/315570520\\_Classification-based\\_RNN\\_machine\\_translation\\_using\\_GRUs](https://www.researchgate.net/publication/315570520_Classification-based_RNN_machine_translation_using_GRUs)
- [5] Tomohiro Fujita, Zhiwei Luo, Changqin Quan, Kohei Mori - Simplification of RNN and Its Performance Evaluation in Machine Translation; 2020  
[https://www.jstage.jst.go.jp/article/iscie/33/10/33\\_267/\\_pdf/-char/en](https://www.jstage.jst.go.jp/article/iscie/33/10/33_267/_pdf/-char/en)
- [6] Sainik Kumar Mahata, Dipankar Das and Sivaji Bandyopadhyay - MTIL2017: Machine Translation Using Recurrent Neural Network on Statistical Machine Translation; 2018  
[https://www.researchgate.net/publication/325456613\\_MTIL2017\\_Machine\\_Translation\\_Using\\_Recurrent\\_Neural\\_Network\\_on\\_Statistical\\_Machine\\_Translation](https://www.researchgate.net/publication/325456613_MTIL2017_Machine_Translation_Using_Recurrent_Neural_Network_on_Statistical_Machine_Translation)
- [7] Zhixing Tan, Shuo Wang, Yang Zonghan, Gang Chen - Neural Machine Translation: A Review of Methods, Resources, and Tools; 2020  
[https://www.researchgate.net/publication/348079690\\_Neural\\_Machine\\_Translation\\_A\\_Review\\_of\\_Methods\\_Resources\\_and\\_Tools](https://www.researchgate.net/publication/348079690_Neural_Machine_Translation_A_Review_of_Methods_Resources_and_Tools)
- [8] Timothy Mayer, Ate Poortinga, Biplov Bhandari, Andrea P. Nicolau, Kel Markert, Nyein Soe Thwal, Amanda Markert, Arjen Haag, John Kilbrideh, Farrukh Chishtie, Amit Wadhwa, Nicholas Clintonj, David Saah - Deep Learning approach for Sentinel-1 Surface Water Mapping leveraging Google Earth Engine; 2021  
[https://www.researchgate.net/publication/355005296\\_Deep\\_Learning\\_approach\\_for\\_Sentinel-1\\_Surface\\_Water\\_Mapping\\_leveraging\\_Google\\_Earth\\_Engine](https://www.researchgate.net/publication/355005296_Deep_Learning_approach_for_Sentinel-1_Surface_Water_Mapping_leveraging_Google_Earth_Engine)
- [9] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu - BLEU: a Method for Automatic Evaluation of Machine Translation; 2002  
[https://www.researchgate.net/publication/2588204\\_BLEU\\_a\\_Method\\_for\\_Automatic\\_Evaluation\\_of\\_Machine\\_Translation](https://www.researchgate.net/publication/2588204_BLEU_a_Method_for_Automatic_Evaluation_of_Machine_Translation)
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin - Attention Is All You Need  
<https://arxiv.org/abs/1706.03762>
- [11] Mani Wadhwa - seq2seq model in Machine Learning, 2021  
<https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning/>