10 марта 2024 г.

1. Введение

Машинный перевод в наши дни достиг высокого качества как по скорости, так и по уровню корректности перевода. Необходимость в автоматическом переводе возникла естественным образом: пользоваться словарем долго и неудобно, необходимо по контексту выбирать наиболее подходящий вариант перевода слова или фразы. При этом необходимо знать на определенном уровне сам язык, его синтаксис и основную лексику. Зачастую человеку необходимо в конкретной ситуации получить перевод пары предложений с определенного языка и, возможно, никогда больше к нему не вернуться.

История машинного перевода начинается с методов, основанных на правилах языка. Модель такого перевода состояла из словаря пары языков и набора правил для каждого языка, например, образование окончаний у частей речи [10]. Минус такого подхода в необходимости привлекать лингвистов и собирать большую базу данных правил языка. В практическом использовании банк правил не сильно помогал в качестве перевода и модели машинного перевода работали "напрямую": сопоставляли слову его перевод по словарю. Незначительно увеличилось качество за счет использования промежуточного языка [8], содержащего обобщенные правила для любых языков. Такой прямой подход позволял понимать содержание текстов, хоть и с грамматическими ошибками, но для похожих языков или языков одной группы. Структура и грамматика японского языка сильно отличается от английского. В 1984 Макото Нагао [5] предложил модель перевода, основанного на фразах, что существенно улучшило качество перевода.

Следующим этапом развития машинного перевода стало использование статистических методов, которые позволили отдалиться от использования языковых правил. Комания IBM предложила статистическую модель [2], основанную на теореме байеса. Если f – исходное предложение, а e – перевод, то распределение возможных переводов задается следующей формулой:

$$P(e|f) = \frac{P(f|e)P(e)}{\sum\limits_{i} P(f|e_i)},$$

где $P(e|f)$ – вероятность перевода e, $P(f|e)$ – вероятность, что исходное предложение f является переводом целевого предложения e. $P(e)$ – вероятность появления данного предложения в корпусе. Наиболее вероятный перевод находится методом максимального правдоподобия при максимизации:

$$e^* = \arg\max_{e} P(f|e)P(e).$$

Фундаментальная проблема в переводе – выравнивание слов. Часто слова в исходном и целевом языках идут в другой последовательности. Многие слова не имеют перевода вовсе, например, артикли при перевода с английского языка на русский. Особенно серьезно эта проблема проявляется, когда один из языков принадлежит азиатской языковой группе. Более подробный обзор статистических методов решения проблемы выравнивания приведен в следующем разделе. Оригинальным решением этой проблемы было представление предложений в виде синтаксического дерева [13], в котором происходит переупорядочивание частей речи под конкретный язык. Как утверждается, этот способ должен был полностью решить проблему выравнивания, однако оказался очень сложным в реализации и не приобрел большой популярности. Наиболее используемым в прикладных задачах стала модель Moses [4], использующая множество подходов статистического перевода и некоторое время была бейзлайном при сравнении с нейронными моделями перевода.

Не смотря на разнообразие методов машинного перевода, проблема выравнивания по прежнему оставалась актуальной. Были трудности с долгосрочными зависимостями в тексте, различным переводом слова или фразы в зависимости от контекста. Авторы статьи [3] предложили использовать воз-

можности обощающей способности реккурентной нейронной сети для кодирования информации исходного текста в вектор скрытого состояния и затем обратного декодирования из одного вектора в целевой текст. Сравнение с Moses по метрике BLEU показало увеличение качества перевода на бенчмарке перевода с английского на французский с 33.30 до 34.54. После такого успеха большинство исследований велось над улучшением архитектур нейронных сетей и над улучшением функций потерь для более эффективного обучения моделей. Для борьбы с затуханием градиентов и забыванием информации в длинных предложениях использовалии различные улучшения, среди которых архитектуры LSTM и GRU. Для ускорения моделей прибегали к использованию сверточных слоев в энкодере и декодере.

Ключевым прорывом в нейронном переводе стала модель трансформер [12], которая сильно улучшила качество кодирования контекстной информации за счет использования слоев self-attention. При значительно меньших ресурсах во время обучения и значительно большой скорости работы модель превзошла на бенчмарке английско-французского перевода предыдущие подходы и метрика BLEU составила 41.0. Трансформерные модели заняли подавляющее большинство ниш использования не только в задаче машинного перевода, но и в других задачах обработки естественного языка: классификация и анализ тональности текстов, суммаризация, генерация и т.д. Трансформеры также распространились и на задачи компьютерного зрения, где также достигали SOTA по основным метрикам.

Нейронный и статистический машинный перевод позволил отказаться от использования знаний о структуре языка и строить модели перевода, имея лишь датасет параллельных текстов на паре языков. Но если для популярных международных языков это большой плюс, то для определенных этнических групп это скорее минус, поскольку собрать такой датасет очень затруднительно и не выгодно коммерчески. Поэтому часть языков не имеет качественного, доступного и быстрого машинного перевода.

В нашей работе мы строим модель русско-башкирского перевода на основе открытого датасета параллельных текстов, собранных из различных исторических или художественных источников с верифицированными переводами.

В качестве модели перевода рассматривается text-to-text transfer transformer (T5) [7], которая представляет собой универсальную енкодер-декодер архитектуру для text-to-text задач. Мы исследуем различные техники обучения для получения наилучшего результата при имеющихся вычислительных ресурсах и данных: transfer learning с задачи multilingual перевода и masked language modeling для предобучения энкодера русских и башкирский текстов. Основной метрикой качества выступает BLEU [6]. Кроме того, мы дополнительно собираем корпус для обучения токенизатора модели в силу его отсутствия для башкирского языка.

2. Русско-башкирский переводчик

2.1. Связанные исследования

Архитектура энкодер-декодер и языковая модель. Развитие машинного перевода началось с использования реккурентных нейронных сетей [3]. Авторы статьи предложили архитектуру энкодер-декодер, которая будет лежать в основе всех последующих генеративных языковых моделей.

Пусть x и y – исходное и целевое преложения соответственно. Тогда вероятность следующего слова y_i при условии, что исходное предложение x и предыдущие слова $y_1, ...y_{i-1}$ задается формулой:

$$P(y_i|y_{i-1}, ..y_1, x) \propto P(y_{i-1}, ...y_1, x|y_i)P(y_i),$$

где $P(y_{i-1}, ...y_1, x|y_i)$ – функция правдоподобия получить данную последовательность слов перевода и его исходного предложения при условии, текущего словаy_i. Таким образом, вероятность следующего слова зависит от предыдущих и входного предложения. По формуле вероятности произведения событий получаем вероятность перевода y при исходном предложении x:

$$P(y|x) = \prod_{i=1}^{T} p(y_i|y_1, y_2, \dots y_{i-1}, x).$$

Данная модель называется авторегрессионной, которая в дальнейшем будет использоваться во всех моделях генерации текста, включая модели перевода.

Эту формулу можно переписать и в другой форме по теореме Байеса. С учетом того, что y_i моделируется параметрической функцией (нейронной сетью) с параметрами (весами) θ, наиболее вероятное слово находится из максимизации функции правдоподобия по параметрам сети:

$$P(y|x) \propto \prod_{i=1}^{T} P(y_{i-1}, ...y_1, x, \theta|y_i)P(y_i) \to \max_{\theta}.$$

$P(y_i)$ – вероятность слова в корпусе текстов и не зависит от θ. Кроме того, можно взять отрицательный логарифм и оптимизировать вместо произведения вероятностей сумму логарифмов:

$$-\sum_{i=1}^{T} \log P(y_{i-1}, ...y_1, x, \theta|y_i) \to \min_{\theta}$$

Как было сказано, y_i моделируется реккурентной нейросетью, состоящей из энкодера и декодера. Базовая модель имела следующий вид. Энкодер задается выражением:

$$h_t = f(W^h h_{i-1} + W^x x_i),$$

где x_t W^h и W^x – матрицы весов скрытого состояния h и входного one-hot вектора x_t. Декодер задается выражением:

$$y_i = f(V^h h_i + V^y y_{i-1}).$$

Схематично архитектура сети представлена на рис. 2.1.

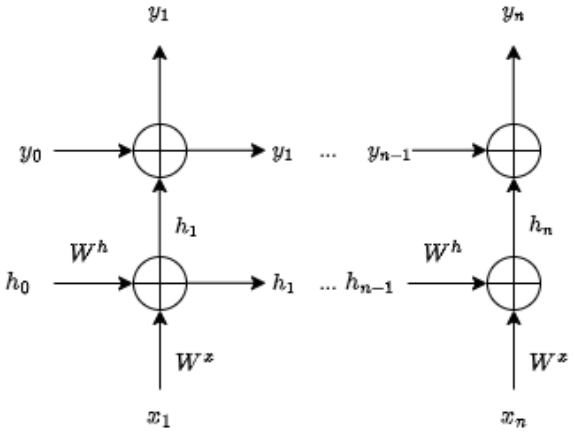

Рис. 2.1. Реккурентная архитектура модели перевода

Заметим, что в такой модели длина исходного и входного предложений должна быть одинакова. На практике такое встречается редко, что является недостатком данного метода. Кроме того, как показывали дальнейшие эксперименты, у такой модели наблюдаются проблемы со сходимостью на длинных текстах. Это было связано с градиентным взрывом или затуханием из-за реккурентной природы данной модели. Модель плохо «запоминает» долгосрочную информацию и контекст теряется. Также подобная архитектура не параллелизуется на GPU по отдельным словам из-за реккурентных слоев. Однако данный метод показал увеличение качества перевода по сравнению с текущим SOTA – Moses [4] с 33.30 до 34.54 по метрике BLEU.

LSTM и последнее скрытое состояние. Следующим шагом в развитии нейронного перевода стала публикация [9]. Авторы предложили способ борьбы с требованием равенства длин входного и целевого предложений. Вместо использования каждого вектора скрытого состояния $h_i, i = \overline{1..T'}$ предлагается использовать только последний h_T – вектор контекста. Авторы полагали, что в нем содержится вся необходимая информация для декодирования перевода (рис. 2.2.

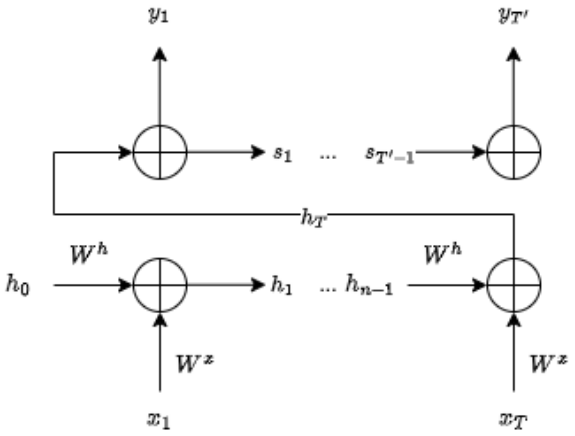

Рис. 2.2. Концепт использования только последнего скрытого состояния

Поскольку размерность целевого предложения переменная, нужны специальные токены, обозначающие начало и конец предложения: <SOS> и <EOS> соответственно. Также авторы увеличили глубину сети до четырех на энкодер и декодер, а в качестве реккурентного слоя использовали LSTM, которая значительно уменьшала проблему взрывных градиентов и забывания

сети. Эмпирическим путем было установлено, что качество генерации перевода лучше, если использовать входное предложение в обратном порядке. На бенчмарке английский-французский метрика BLEU составила 34.81.

Механизм внимания. Не смотря на уменьшение «забывания» сети и решение проблемы равной длины входных и выходных предложений, по прежнему оставалась ярко выражена проблема выравнивания слов. Вектор последнего скрытого состояния не может содержать в себе полную информацию о контексте всего предложения и взаимосвязи слов исходного предложения и целевого. Авторы статьи [1] утверждают, что последее скрытое состояние h_T содержит недостаточно информации для генерации предложения. Часть признаков теряется, которая изначально содержалась в предыдущих скрытых состояниях. В h_i содержится информация, которая характеризует взаимосвязь со словом из перевода. Авторы предложили вычислять степень связи исходного слова, выраженного скрытым состоянием h_i, и перевода s_i с помощью некоторой меры схожести $sim(s_i, h_i)$. В оригинальной статье используется небольшая полносвязная нейросеть, но может быть использована любая мера схожести, например, косинусная. При этом авторы испольуют меру схожести, нормированную на сумму схожестей перевода s_i со всеми скрытыми состояниями h_i:

$$\alpha_{ij} = \frac{sim(s_i, h_i)}{\sum\limits_{k=1}^{T} sim(s_i, h_k)}.$$

Логично предположить, что слово-перевод связано не с единственным словом из исходного предложения. Для этого авторы предлагают взвесить вклад каждого слова из исходного предложения в перевод:

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j.$$

Если посмотреть на формулу, то можно провести параллель с условным математическим ожиданием. В данном случае условие – это слово-перевод s_i, а коэффициенты α_{ij} – вероятность того, что скрытое представление h_j вносит свой вклад в перевод (связано с переводом). Данный вектор назвали вектором контекста, а сам механизм выравнивания слов – механизмом внимания.

Слой декодера с учетом вектора контекста принял вид:

$$s_i = f(W^s_{i-1} + W^c c_i),\ i = \overline{1, T'}.$$

Данная модель превзошла предыдущие бейзлайны по качеству перевода на основных бенчмарках. Вектор контекста значительно повлиял на проблему выравнивания слов, а механизм внимания стал основной концепцией в трансформерных архитектурах и нашел применение практически во всех модальностях данных, включая компьютерное зрение и обработку звука.

Трансформер и self-attention. Прорыв в обработке последовательностей с использованием механизмов внимания внес бльшой влкад в метрики качества моделей, но вычислительные проблемы по прежнему оставались: плохая параллелизуемость реккурентных нейросетей, градиентный взрыв и затухание. Вдохновляясь работой [1] авторы статьи [11] предложили отказаться от реккурентной архитектуры, а все слои нейросети реализовать в виде «слоев внимания» и предложили новую архитектуру – трансформер. Он также использует концепцию энкодер-декодер. Модель генерации перевода не изменяется. Помимо классического механизма внимания, авторы предалагют вычислять связь между словами входного предложения. Мотивация заключается в том, что перевод зависит от контекста употребления различных слов или фраз, при этом контекст может быть не только локальным, но и глобальным.

Чтобы измерять степень связи слов исходного предложения авторы ввели self-attention слой:

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

где Q, K, V – матрицы «запроса», «ключа» и «значения». d_k – число столбцов матрицы Q.

Чтбы лучше понять эту формулу, посмотрим на первый слой энкодера. Пусть X исходная последовательность токенов. Токен представлен в виде вектора (эмбеддинга), следовательно, X – матрица размерности $\mathbb{R}^{n \times d_{model}}$, где n – длина последовательности, а $model$ – размерность эмбеддинга. Отобразим линейно каждый эмбеддинг в пространство размерности k и v – это

нужно для большей «гибкости» – с помощью матриц $W_Q \in \mathbb{R}^{d_{model} \times d_k}, W_K \in \mathbb{R}^{d_{model} \times d_k}, W_V \in \mathbb{R}^{d_{model} \times d_v}$. Мы получим три новых матрицы $Q = XW_Q, K = XW_K, V = XW_V$. Далее мы можем вычислить новый вектор скрытого состояния.

Приведем интерпретацию операций в self-attention слое. Предположим, что матрицы W_i – единичные, то формула получится следующей:

$$attention(Q, K, V) = \text{softmax}(\frac{XX^T}{\sqrt{d_{model}}})X.$$

В числителе XX^T – матрица Грама или нецентрированная матрица ковариации. Ковариация характеризует степень связи двух случайных величин, в нашем случае двух слов. Вспомним, что i-я и j-я строка – эмбеддинги токенов i и j. Тогда матрица XX^T будет характеризовать то, как связано слово i со словом j. Соответственно каждая строка i данной матрицы – набор «коэффициентов связи» слова i с остальными словами в предложении, нормированная на коэффициент $\sqrt{d_k}$.

Пусть x_i – вектор-строка матрицы X. Для простоты выкладок опустим операцию softmax и коэффициент масштабирования. Тогда операция self-attention слоя выражается следующим образом:

$$XX^TX = \begin{pmatrix} x_1x_1^T & \dots & x_1x_n^T \\ \dots & \dots & \dots \\ x_nx_1^T & \dots & x_nx_n^T \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} \left(x_1x_i^T\right) x_i \\ \dots \\ \sum_{i=1}^{n} \left(x_nx_i^T\right) x_i \end{pmatrix}.$$

Заметим, что полученное выражение – матрица, строки которой представляют собой взвешенную сумму эмбеддингов x_i. При этом вес слагаемого в i-й строке – скалярное произведение, характиризующее силу связи i-го слова со всеми остальными словами. После первого слоя мы будем иметь матрицу, содержающую информацию о попарных связях между токенами. В следующем слое – связь между би-граммами и так далее.

Масштабирующий коэффициент $\sqrt{d_k}$. Матрица ковариаций состоит из ненормированных значений и при увеличении размерности случайных величин их ковариация будет увеличиваться или уменьшаться. В нейросетях

это может привести к взрыву градиентов. Свойством нормировки обладает матрица корреляций, которая получается путем деления ковариации на стандартные отклонения соответствующих случайных величин. В теории, можно было получить из матрицы ковариаций корреляционную матрицу, но это могло увеличить вычислительные затраты. Хорошим решением на практике стало нормирование данной матрицы на размерность случайной величины. Также сами авторы приводят пример с тем, что происходит с дисперсией скалярного произведения двух независимых случайных величин с дисперсией 1:

$$qk = \sum_{i=1}^{d} q_i k_i.$$

В реккурентных моделях генерация токенов последовательно получалась естественным образом и модель не могла «подглядывать» на следующие токены в целевом предложении. В трансформере эту проблему предложили решить с помощью маскирующего слоя: при генерации y_i токена $\overline{i+1, T}$ токены домножаются на нулевые веса. Также авторы предлагают способ закодировать информацию о позиции слова в предложении, так как предложенная архитектура не учитывает порядок слов. Подход был назван positional encoding и позволяет выучивать трансформеру относительные позиции слов в предложении. Модель превзошла существующие подходы по метрике BLEU в переводах english-german (28.4) и english-french (41.8) и по вычислительной эффективности ($3.3 \cdot 10^{18}$) FLOPS. Работа стала прорывом в обработке текстов и большинство последующих моделей используют данную архитектуру в качестве основного блока архитектуры нейросети.

Byte-pair-encoding.

Sentencepiece.

T5.

3. Датасет

В качестве датасета был взял открытый корпус параллельных русско-башкирских текстов[1]. Датасет содержит 712 тысяч обучающих примеров. Параллельные тексты были составлены из переводов книг, новостей, переводов профессионального переводчика, энциклопедий[2]. Максимальная длина среди русских и башкирских предложений 424 слова. Распределение по длинам изображено на рис. 3.1. Исходя из информации о длине предложений

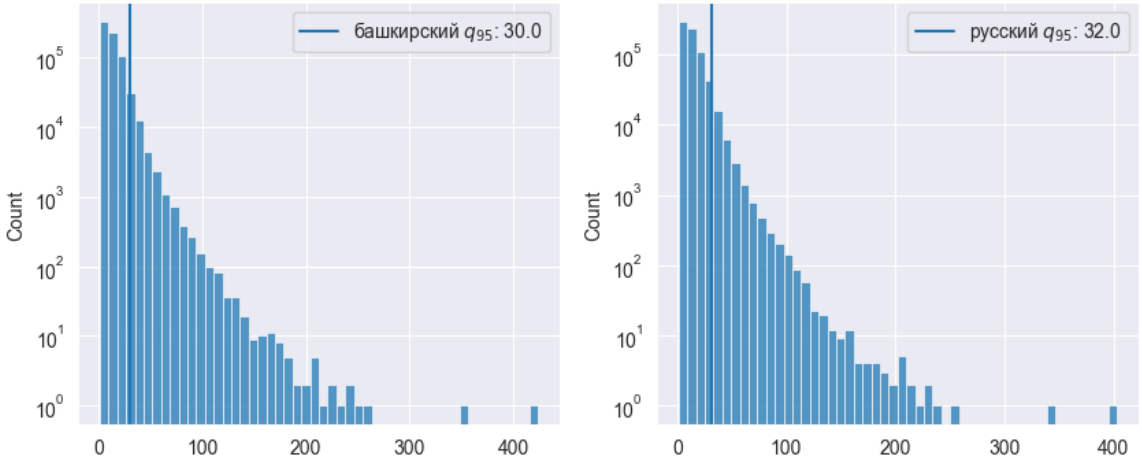

Рис. 3.1. Распределение длин предложений с указанием 95-го квантиля

при экспериментах можно ограничиться максимальной размерностью входа кратного двум, в данном случае 512. Также проводятся эксперименты с исключением прежлложений длинее 32-х слов (95-й квантиль). Пример данных представлены в таблице 1.

Таблица 1. Примеры текстов датасета

русский	**башкирский**
Мне вдруг захотелось заплакать	Илағым килде ҡапыл
Волк ударил моего старика	Бүре һуҡты, бабайҙы бүре һуҡты!

[1] https://huggingface.co/datasets/AigizK/bashkir-russian-parallel-corpora

[2] https://github.com/AigizK/bashkort-parallel-corpora

4. Модель

T5. Модель T5 позволяет решать множество задач одновременно. Авторы позиционировали данную нейросеть как универсальную языковую модель. В ее основе лежит трансформерная энкодер-декодер архитектура. Ключевая особенность модели лежит в способе обучения. Авторы предлагают два режима: masked language modeling (MLM) в случае обучения без учителя и в режиме text-to-text. При этом в обоих вариантах модель обучается генерации текста. В случае MLM модель учится генерировать пропущенную часть текста на входном предложении, а в случае text-to-text – генерировать целевые последовательности. Первый вариант обучения – аналог модели BERT и используется для downstream задач, например, для классификации текстов.

Наша задача машинного перевода – частный случай задачи генерации целевого текста из исходного, поэтому мы используем второй режим обучения. При этом в экспериментальную часть также войдут попытки предобучения модели методом MLM с целью улучшить дообучения text-to-text. В процессе обучения мы также будем применять технику teacher forcing для улучшения сходимости модели. Поскольку наша задача получить переводчик как с русского на башкирский, так и с башкирского на русский, то логичный шаг – обучать модель одновременно двум языкам. Для определения целевого языка перевода в процессе обучения и инференса используются специальный токен: <SOURCE_LANG>-<TARGET_LANG>, где SOURCE_LANG и TARGET_LANG могут принимать значения BAK – башкирский и RU – русский. <TODO: ОНФИГУРАЦИЯ МОДЕЛИ ВСТАВИТЬ>

Токенизатор. В модели T5 используется sentencepiece токенизатор <TODO НАПИСАТЬ ПОДРОБНЕЕ>. Токенизаторов для башкирского языка нет в открытом доступе, при этом мы также не можем воспользоваться только токенизатором русского текста в силу большого различия в языках. Мы обучаем новый токенизатор на объединенных башкирских и русских текстов, образуя общий словарь символов. Помимо обучающего набора параллельных текстов датасет для токенизатора был расширен русскими тексты из интернета, башкирскими худождественными книгами и новостями. Размер словаря

был выбран 32000 токенов.

Базовая конфигурация экспериментов. Используется оптимизатор Adam с начальной скоростью обучения 0.001 с сокращением скорости обучения в 10 раз при выходе функции потерь на плато и отсутствии улучшения в метриках в течение 10 эпох. Ранняя остановка после 50ти эпох отсутствия улучшения в метриках. Функция потерь кросс-энтропия.

Тестирование и бейзлайн. В качестве базовой модели был выбран сервис яндекс переводчик. Из тестового набора данных было отобрано по 300 примеров для перевода с русского на башкирский и с башкирского на русский. В качестве основной метрики используется BLEU.

5. Эксперименты

Корректность датасета. Для проверки валидности датасета были проведены тестовые запуски обучения на различном размере обучающего набора. Цель данного эксперимента – убедиться, что при увеличении обучающего набора качество модели, т.е. метрика BLEU будет увеличиваться. Модель обучалась в течение 10-и эпох на размере датасета 10k, 50k, 200k; 1M. Отметим, что фактический размер датасета увеличился в два раза, поскольку перевод осуществляется одновременно на два языка. Таким образом, если в исходном датасете имеется пара текстов source-target, то во время обучения будет две пары: source-target и target-source. График тестовых обучений приведен на рис. 5.1. Как можно видеть, BLEU увеличивается с увеличением размера обучающей выборки, следовательно, можно следать вывод о корректности данных. Также для полного набора данных (1M) приведен график уменьшения функций потерь на валидационной выборке (рис. 5.2). Отметим, что на 10-й эпохе наблюдается начало расхождения функций потерь на обучении и валидации, что может свидетельствовать о начале переобучения. В последующих экспериментах следует обратить внимание на число параметров модели, а также регуляризацию. Первичные результаты сравнения нашей модели с бейзлайном представлены в таблице 2.

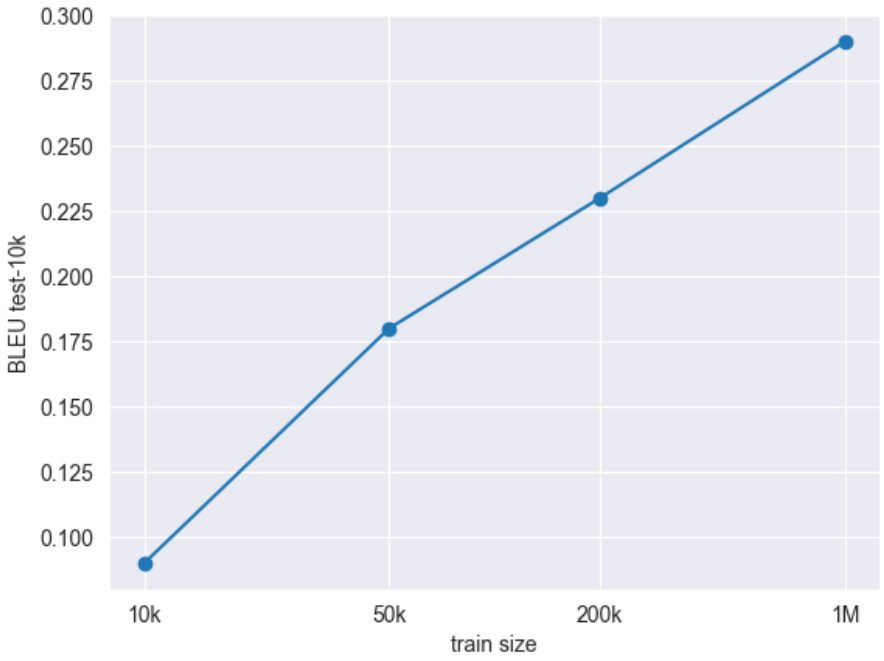

Рис. 5.1. Изменение BLEU в зависимости от размера датасета

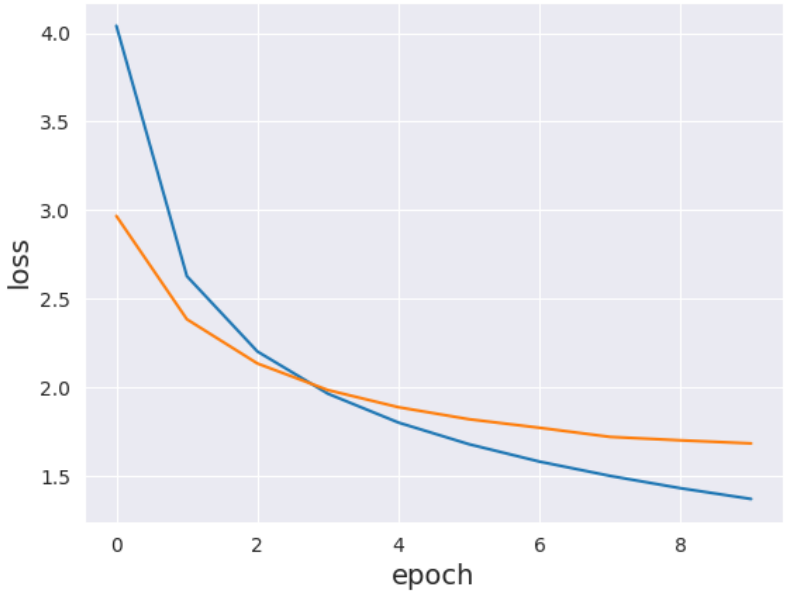

Рис. 5.2. Функции потерь на полном размере обучающего множества

Таблица 2. Сравнение первичных результатов с бейзлайном

Модель	ru-bak	bak-ru
yandex translate	**0.49**	**0.53**
t5 base	0.29	0.31

6. Планы дальше

- Попробовать предобучить seq2seq через MLM
- Расширить токенизатор
- Менять размер модели
- Что еще?

Список литературы

[1] Dzmitry Bahdanau, Kyunghyun Cho и Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL].

[2] Peter F Brown и др. "The mathematics of statistical machine translation: Parameter estimation". В: (1993).

[3] Kyunghyun Cho и др. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: 1406.1078 [cs.CL].

[4] Philipp Koehn и др. "Moses: Open source toolkit for statistical machine translation". В: *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*. Association for Computational Linguistics. 2007, с. 177—180.

[5] Makoto Nagao. "A framework of a mechanical translation between Japanese and English by analogy principle". В: *Artificial and human intelligence* (1984), с. 351—354.

[6] Kishore Papineni и др. "Bleu: a method for automatic evaluation of machine translation". В: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, с. 311—318.

[7] Colin Raffel и др. "Exploring the limits of transfer learning with a unified text-to-text transformer". В: *The Journal of Machine Learning Research* 21.1 (2020), с. 5485—5551.

[8] Richard H Richens. "Interlingual machine translation". В: *The Computer Journal* 1.3 (1958), с. 144—147.

[9] Ilya Sutskever, Oriol Vinyals и Quoc V Le. "Sequence to sequence learning with neural networks". В: *Advances in neural information processing systems* 27 (2014).

[10] Peter Toma. “Systran as a multilingual machine translation system”. B: *Proceedings of the Third European Congress on Information Systems and Networks, overcoming the language barrier*. 1977, с. 569—581.

[11] Ashish Vaswani и др. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL].

[12] Ashish Vaswani и др. “Attention is all you need”. B: *Advances in neural information processing systems* 30 (2017).

[13] Kenji Yamada и Kevin Knight. “A syntax-based statistical translation model”. B: *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*. 2001, с. 523—530.