

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
"МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ  
(ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ)"

ФАКУЛЬТЕТ ИННОВАЦИЙ И ВЫСОКИХ ТЕХНОЛОГИЙ  
КАФЕДРА ДИСКРЕТНОЙ МАТЕМАТИКИ

---

Выпускная квалификационная работа по  
направлению 01.03.02 «Прикладные математика и  
информатика»  
НА ТЕМУ:

**ДИФФЕРЕНЦИРУЕМАЯ НИЖНЯЯ ГРАНИЦА ДЛЯ  
МАТЕМАТИЧЕСКОГО ОЖИДАНИЯ МЕТРИКИ BLEU.**

Студент \_\_\_\_\_ Жуков В.А.

Научный руководитель к.ф.-м.н. \_\_\_\_\_ Кретов М.К.

МОСКВА, 2018

# Содержание

<b>1</b>	<b>Введение и основные понятия</b>	<b>4</b>
1.1	Задача машинного перевода . . . . .	4
1.2	Метрики качества . . . . .	4
1.3	Sequence-to-sequence архитектура . . . . .	4
1.3.1	Лучевой поиск в sequence-to-sequence модели . . . . .	6
1.3.2	Механизм внимания в sequence-to-sequence модели . . . . .	6
1.4	Обучение с подкреплением и правило REINFORCE . . . . .	8
1.5	Стохастические вычислительные графы . . . . .	10
<b>2</b>	<b>Цель работы</b>	<b>12</b>
<b>3</b>	<b>Постановка задачи и используемые обозначения</b>	<b>13</b>
3.1	Метрика BLEU . . . . .	13
3.1.1	Пример расчета метрики BLEU . . . . .	14
<b>4</b>	<b>Предлагаемый метод</b>	<b>15</b>
4.1	BLEU в матричном виде . . . . .	15
4.2	Нижняя граница математического ожидания метрики BLEU . . . . .	16
<b>5</b>	<b>Эксперименты и результаты</b>	<b>19</b>
5.1	Модельная задача . . . . .	19
5.2	Результаты в задаче перевода . . . . .	20
5.2.1	Используемая модель . . . . .	22
5.2.2	Детали обучения модели . . . . .	23
<b>6</b>	<b>Выводы</b>	<b>24</b>
	<b>Приложения</b>	<b>27</b>
<b>A</b>	<b>Пример расчета метрики BLEU в матричном виде</b>	<b>27</b>
<b>B</b>	<b>Эквивалентность формул для перекрытий <math>O_n</math></b>	<b>28</b>
B.1	Пример . . . . .	28
<b>C</b>	<b>Нижняя граница для униграмм</b>	<b>29</b>

D	Обобщение для n-грамм	29
E	Доказательство выпуклости	31
F	Полные результаты экспериментов на датасете IWSLT	31

## Аннотация

В задачах обработки естественного языка качество модели обычно измеряется некоторой недифференцируемой метрикой, такой как, например, BLEU [1]. Для того, чтобы использовать имеющиеся эффективные градиентные методы оптимизации, обычно используют некоторую суррогатную дифференцируемую функцию потерь. Этот подход эффективен, если при оптимизации такой функции потерь улучшается и целевая метрика. В этой работе предлагается метод для вычисления дифференцируемой нижней границы для метрики BLEU. А также приводятся результаты ее работы как на модельной задаче, так и на задаче машинного перевода. Новый метод не включает в себя вычислительно дорогую процедуру сэмплирования, которая требуется, например, при использовании метода REINFORCE из обучения с подкреплением.

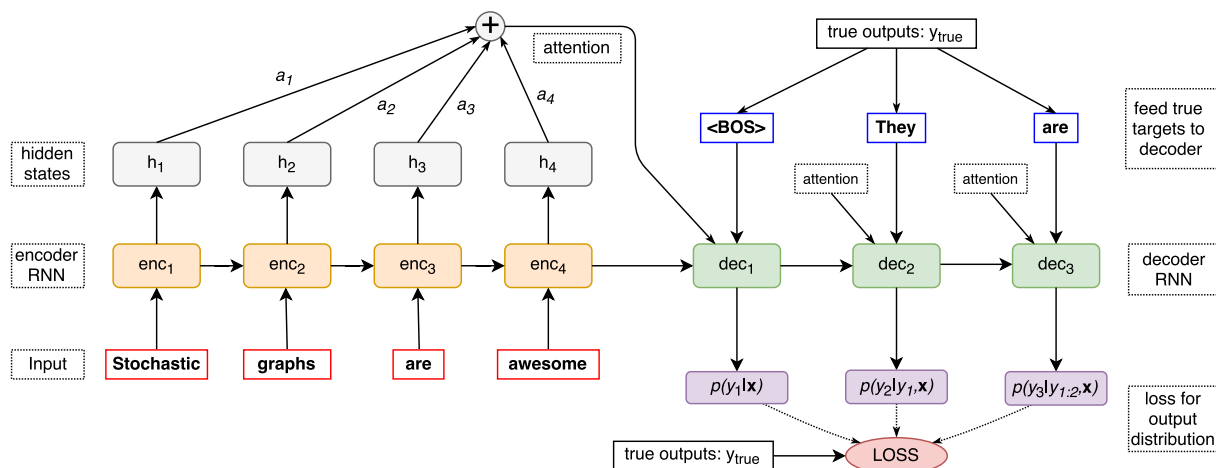


Рис. 1: Пример модели seq2seq с механизмом внимания; фаза обучения

# 1 Введение и основные понятия

## 1.1 Задача машинного перевода

Машинный перевод - это подраздел компьютерной лингвистики, который изучает вычислительные методы перевода с одного языка на другой. На вход алгоритму машинного перевода дается текст на исходном языке, а задачей является предсказать текст на другом языке.

## 1.2 Метрики качества

Одной из особенностей задач обработки естественного языка являются сложные метрики качества. Часто тяжело работать с такими метриками в абсолютно строгих терминах. Например, довольно трудно дать количественную оценку качеству текста также, как его оценил бы человек. Поэтому используют контрольные метрики: BLEU [1], ROUGE [2]. Эти метрики не решают вышеупомянутую проблему полностью [3], но они интерпретируемы и легко вычислимы.

## 1.3 Sequence-to-sequence архитектура

Sequence-to-sequence модели (seq2seq [4]) - это широкий класс моделей, которые предсказывают выходную последовательность из некоторой входной последовательности. Такая архитектура применяется во многих за-

дачах обработки естественного языка, например таких, как машинный перевод [5] и суммаризация [6].

На рисунке 1 показан типичный пример seq2seq модели, состоящей из кодировщика(encoder), задача которого получить внутреннее представление входной последовательности, и декодера(decoder), задача которого по этому представлению получить желаемую выходную последовательность. В данном примере в качестве входной последовательности на вход кодировщику подается последовательность “Stochastic graphs are awesome”. Кодировщик последовательно считывает последовательность, получая на каждом шаге скрытое представление  $h_i$ .

Декодер имеет внутреннее состояние, которое инициализируется представлением последовательности(последним скрытым состоянием), полученным из кодировщика. Далее декодер на каждом шаге предсказывает условное распределение на текущий элемент последовательности при условии входной последовательности и предыдущих элементов последовательности. При подходе teacher forcing [7] в качестве предыдущих элементов последовательности берутся слова из целевого предложения. Это обеспечивается тем, что на каждом шаге декодер получает на вход предыдущее слово из целевого предложения. На рисунке 1 также показано как во время начала декодирования декодер получает на вход слово, обозначающее начало предложения – <BOS>. Помимо слова из целевого предложения декодер получает “контекстный вектор”, полученный с помощью механизма внимания, который будет описан ниже в разделе 1.3.2.

Одним из подходов к тренировке таких моделей является максимизация правдоподобия истинной последовательности при условии входной последовательности и предыдущих истинных элементов последовательности. Такой подход известен как teacher forcing [7]. Несмотря на то, что метод достаточно эффективен на практике, во время вывода(тестирования) модель не имеет доступа к истинным элементам последовательности и может использовать только свои собственные предсказания. Таким образом, работа модели во время тренировки и тестирования отличаются. Это приводит к двум главным проблемам:

- Exposure bias [8] - модель не использует свои собственные предсказания во время тренировки
- Loss-evaluation mismatch [9] - во время тренировки мы оптимизируем дифференцируемую метрику, но меряем качество другой метрикой.

### 1.3.1 Лучевой поиск в sequence-to-sequence модели

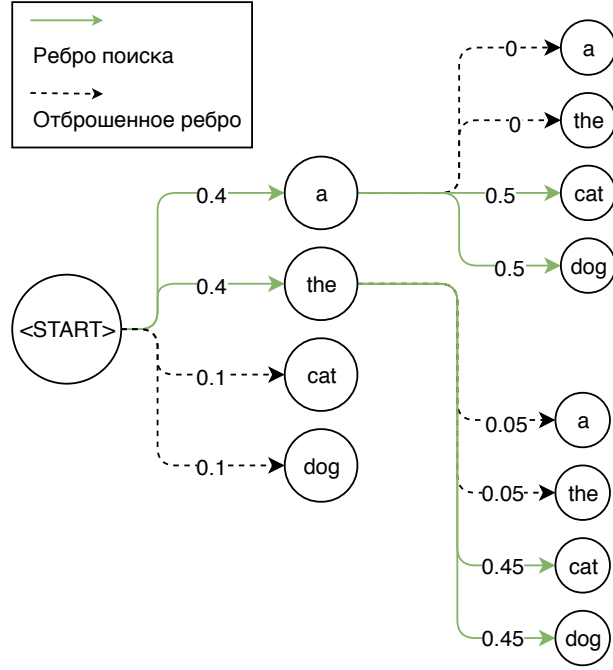
Лучевой поиск (beam search) - это эвристический жадный алгоритм поиска пути в графе. По своей сути он модифицирует поиск по наилучшему совпадению. Для каждого состояния вводится функция оценки. Во время работы алгоритма поддерживается множество лучших состояний с точки зрения функции оценки. Каждый шаг алгоритма представляет собой несколько этапов:

- Сделать шаг из множества лучших состояний.
- Оставить только уникальные состояния, удалив лишние состояния-дубликаты.
- Посчитать функцию оценки для текущих состояний и выбрать по ней *beamsize* лучших.

В задаче генерации текста алгоритм может быть применен к графу, вершинами которого являются слова, а веса ребер из каждой вершины задаются условным распределением на словах (вероятность сгенерировать слово при условии предыдущих). Каждому состоянию будет соответствовать последовательность слов. Оценкой для состояния будет сумма логарифмов вероятностей слов выданных декодером. Алгоритм продемонстрирован на рисунке 2 (в данном примере *beamsize* = 2).

### 1.3.2 Механизм внимания в sequence-to-sequence модели

Проблема seq2seq моделей (без механизма внимания) в том, что кодировщик должен представить всю входную последовательность в виде вектора конечной размерности. Механизм внимания или просто внимание —



**Рис. 2:** Пример лучевого поиска осуществляемого декодером. Числа на ребрах - это вероятности сгенерировать слово.

это метод, который часто используется в seq2seq моделях для решения этой проблемы. Он может быть использован для того, чтобы во время декодирования обращаться напрямую к:

1. Состояниям кодировщика.
2. Предыдущим состояниям декодера.

В данной работе используется первый вариант. Используя внимание можно получить так называемый контекстный вектор как взвешенное среднее скрытых состояний кодировщика  $e_1 \dots e_m$

$$c_i = \sum_j a_{i,j} e_j$$

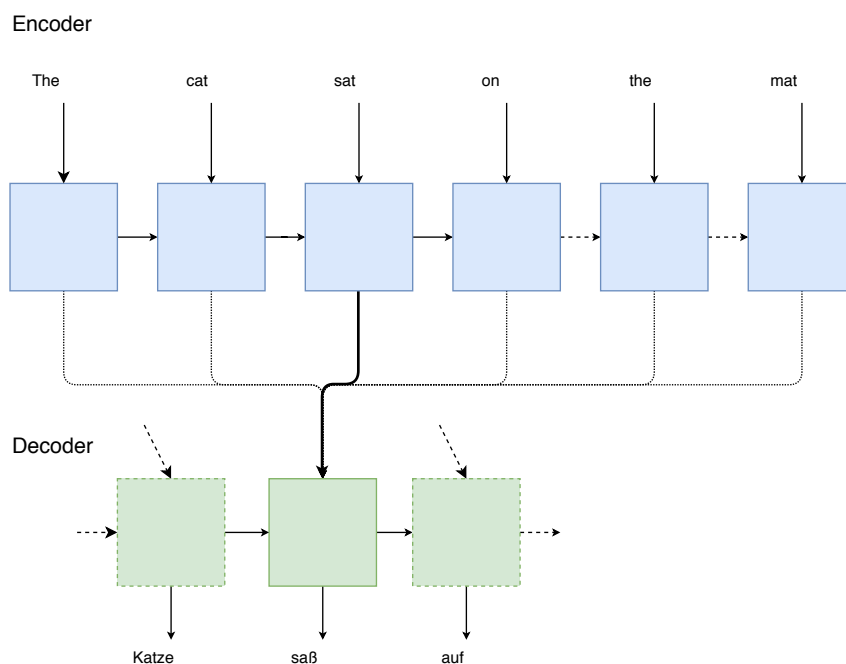
$$a_i = \text{softmax}(f_{att}(h_i, e_j))$$

Алгоритм продемонстрирован на рисунке 3. Функция  $f_{att}(h_i, e_j)$  вычисляет выравнивание(alignment) между скрытыми состояниями  $h_i$  и  $e_j$ . В работе используется мультипликативное внимание. При таком подходе  $f_{att}$  вычисляется как:

$$f_{att} = h_i^T W_a e_j$$

$W_a$  – это обучаемые параметры внимания.

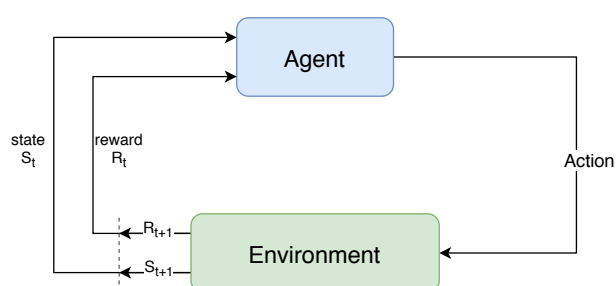




**Рис. 3:** Иллюстрация обученного механизма внимания. Во время генерации слова “saß”, с большим весом берется состояние кодировщика соответствующего моменту времени, когда на вход подавалось слово “sat”.

## 1.4 Обучение с подкреплением и правило REINFORCE

Несмотря на то, что терминология обучения с подкреплением в задаче оптимизации недифференцируемой метрики не является необходимой [10], приведем здесь описание правила REINFORCE именно в этих терминах.



**Рис. 4:** Взаимодействие агента со средой в Марковском процессе принятия решений

Марковский процесс принятия решений (MDP — Markov decision process) — это процесс в котором участвуют агент и среда. Процесс подразумевает собой взаимодействие среды и агента. Это взаимодействие происходит последовательно и изображено на рисунке 4. Агент — это некоторый алгоритм, который в каждый момент времени в ходе марковского процесса

принятия решений выбирает действие. Среда в соответствующий момент времени реагирует на это действие, и агент оказывается в новом состоянии (среда дает новое состояние агенту). Помимо состояния среда дает агенту награду – значение, которое пытается максимизировать агент в ходе взаимодействия со средой.

Более формально (конечный марковский процесс принятия решений):

- Состояние  $S_t \in \mathcal{S}$  – это представление среды в момент времени  $t \in \{0, 1, 2, 3, \dots\}$ , полученное агентом.
- В момент времени  $t$  агент может выбрать действие  $A_t \in \mathcal{A}$ .
- После того как агент выбрал действие, он получает награду  $R_t \in \mathcal{R} \subset \mathbb{R}$  и оказывается в состоянии  $S_{t+1}$

Для принятия решений агенты используют так называемую политику. Политика  $\pi$  – это отображение из  $\mathcal{S}$  в распределение на возможных действиях для данного состояния. Если это распределение вырожденное для всех состояний, то такую политику называют детерминистической, иначе стохастической.

Траектория – это последовательность состояний и действий (иногда в траекторию также включают и награды)

$$\tau = S_0 A_0 (R_1) S_1 A_1 (R_2) \dots$$

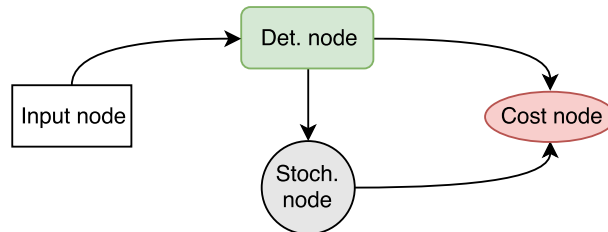
Траекторию можно задать также относительно политики  $\pi$  (обозначение  $\tau_\pi$ ), тогда в каждый момент времени  $t$  действием будет являться случайная величина порожденная распределением  $\pi_t(S_t)$ .

Общим подходом для оптимизации недифференцируемых метрик является использование методов из обучения с подкреплением (RL – reinforcement learning), например, REINFORCE, который позволяет оптимизировать математическое ожидание нужной метрики.

$$\begin{aligned} J(\theta) &= \mathbb{E}_{\tau \sim \pi_\theta} [R(\tau)] \\ \Rightarrow \nabla_\theta J(\theta) &= \mathbb{E}_\tau \left[ \sum_t R_t \nabla_\theta \log \pi_\theta(a_t | s_t) \right] \end{aligned} \quad (1)$$

Здесь  $R(\tau)$  это общая награда полученная агентом во время прохождения траектории  $\tau$ :  $R(\tau) = \sum_t R_t$ . В контексте задач обработки естественного языка траектория может рассматриваться как последовательность слов сгенерированных декодером и скрытых состояний декодера  $(w_1, h_1, ..w_N, h_N)$ , а в роли награды  $R(\tau)$  может выступить метрика BLEU. Стохастическая политика  $\pi_\theta(a_t|s_t)$  задается распределением на словах в момент времени  $t$ . У такого подхода есть проблема – интегрируемая случайная величина имеет довольно большую дисперсию, что делает затруднительным получение численных оценок этого интеграла. Одним из методов борьбы с этой проблемой является подбор baseline алгоритма. Тогда вместо  $R(\tau)$  используют функцию  $A(\tau) = R(\tau) - R_B$ . Где  $R_B$  - это награда полученная baseline алгоритмом. В RL в качестве baseline алгоритма используется value функция  $V(S)$  – математическое ожидание награды для данного состояния.

## 1.5 Стохастические вычислительные графы



**Рис. 5:** Пример стохастического вычислительного графа

Стохастический вычислительный граф [11] - это направленный ациклический граф с тремя типами вершин:

- Входные вершины
- Детерминистические вершины – это функции от родительских вершин
- Стохастические вершины – это распределения при условии родительских вершин.

В статье [11] доказывается следующая теорема:

**Теорема 1.** Примем следующие обозначения:

- $\Theta$  - множество входных вершин.
- $\mathcal{C}$  - множество вершин “стоимости”(cost)
- $\mathcal{S}$  - множество стохастических вершин.
- $u \prec v$  - обозначает что существует путь из  $u$  в  $v$ .
- $u \prec^D v$  - обозначает что существует путь из  $u$  в  $v$  все вершины которого за исключением последней являются детерминистическими.
- $\hat{c}$  - реализация вершины “стоимости”  $c$
- $\text{DEPS}_v = \{w \in \Theta \cup \mathcal{S} | w \prec^D v\}$ .

Пусть для  $\theta \in \Theta$  выполняются условия на дифференцируемость [11], тогда

$$\frac{\partial}{\partial \theta} \mathbb{E}_{v \in \mathcal{S}} \left[ \sum_{c \in \mathcal{C}} c \right] = \mathbb{E}_{v \in \mathcal{S}} \left[ \underbrace{\sum_{w \in \mathcal{S}, \theta \prec^D w} \left( \frac{\partial}{\partial \theta} \log p(w | \text{DEPS}_w) \right) \sum_{c \in \mathcal{C}, w \prec c} \hat{c}}_{(A)} + \underbrace{\sum_{c \in \mathcal{C}, \theta \prec^D c} \frac{\partial}{\partial \theta} c(\text{DEPS}_c)}_{(B)} \right].$$

Формула состоит из двух частей:

(B) - соответствует обычному алгоритму обратного распространения ошибок.

(A) - отвечает стохастическим узлам.

Эта теорема позволяет вычислять градиенты функции “стоимости” по параметрам стохастического вычислительного графа.

Вспомним две главные проблемы тренировки модели seq2seq:

- Exposure bias. Модель не использует свои собственные предсказания во время тренировки.

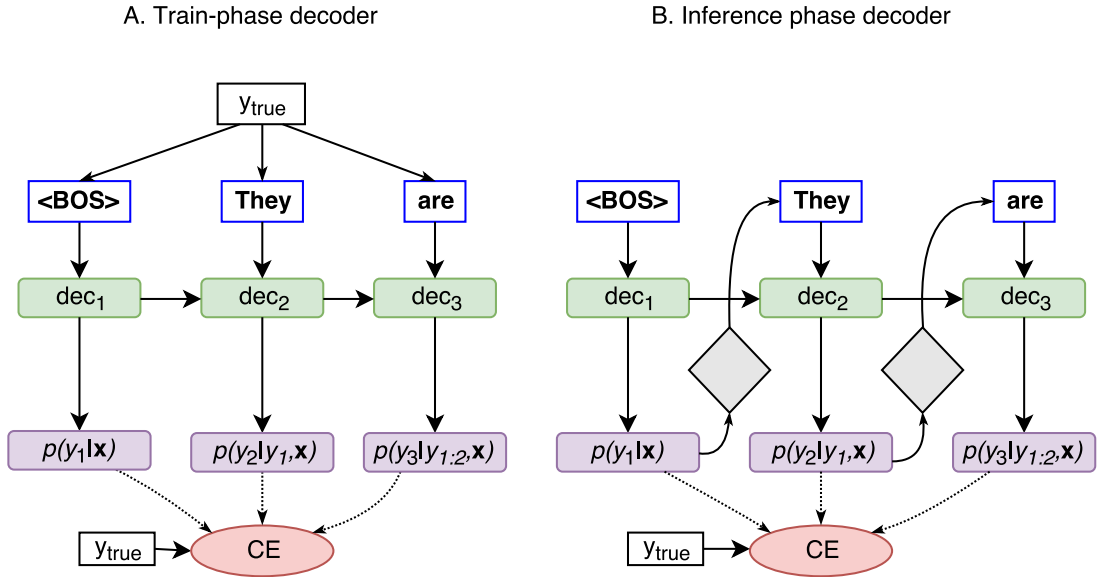


Рис. 6: Декодер во время стадий тренировки и применения

- Loss-evaluation mismatch. Во время тренировки мы оптимизируем дифференцируемую метрику, но меряем качество другой метрикой (например BLEU)

Для того, чтобы избавиться от первой проблемы, мы могли бы подавать на вход декодеру реализацию случайной величины с распределением, которые он предсказал на предыдущем шаге [12]. Изменив таким образом модель, мы вводим случайность в процесс обучения и наш вычислительный граф становится стохастическим (рис. 6B).

Для решения второй проблемы можно вычислять ответы сэмплингом и распределений которые выдает декодер.

## 2 Цель работы

Здесь и далее будем обозначать значение метрики для конкретных предложений  $x$  и  $y$  как  $\text{BLEU}(x, y)$ .

Задачу максимизации математического ожидания метрики BLEU можно представить в следующем виде

$$\mathbb{E}_{\text{data}} \mathbb{E}_{\text{arch\_stoch}} \mathbb{E}_{\text{output}} \text{BLEU}(x, y) \rightarrow \max, \quad (2)$$

Первое математическое ожидание берется по тренировочной выборке. Второе берется по “стохастичности сети”, которая имеет место быть ес-

ли тренируемая сеть является стохастическим вычислительным графом [11]. Третье вызвано тем, что декодер предсказывает распределение на словах(элементах последовательности), а не сами слова. В данной работе рассматривается именно третье математическое ожидание. Общим подходом для его оптимизации является формула 1, однако интеграл в этой формуле оказывается вычислительно трудным. В данной работе выводится аналитическая формула дифференцируемой нижней границы метрики BLEU (BLB). Ее эффективность показывается как на модельной задаче оптимизации, так и в задаче машинного перевода, где seq2seq модель дообученная с помощью BLB в среднем показывает лучшее качество чем модель дообученная классическим методом REINFORCE.

## 3 Постановка задачи и используемые обозначения

### 3.1 Метрика BLEU

Метрика BLEU является одной из самых популярных метрик для оценки качества моделей генерации текста и в частности машинного перевода. Как и все подобные метрики она зависит от двух аргументов: предложения гипотезы  $C$ (candidate) и истинного предложения  $R$ (reference). В данной работе мы не будем рассматривать метрику BLEU для корпуса и будем использовать усреднение (итоговая метрика BLEU указывается для корпуса). Вычисление этой метрики состоит из вычисления модифицированных точностей  $p_n$  по  $n$ -граммам и штрафа за краткость BP(brevity penalty). Модифицированные точности считаются следующим образом

$$p_n = \frac{O_n}{\sum_{C \in \{Candidates\}} \sum_{n\text{-gram} \in C} \text{Count}(n\text{-gram})} \quad (3)$$

Знаменатель этой дроби является ни чем иным как количеством возможных  $n$ -грамм в гипотезе  $C$ . Числитель состоит из так называемых “перекрытий”  $O_n$ (overlaps) между гипотезой  $C$  и истинным предложением.

ем  $R$  и рассчитывается как:

$$\sum_{n\text{-gram} \in C} \min(\text{Count}_C(n\text{-gram}), \text{Count}_R(n\text{-gram})) \quad (4)$$

Из формулы видно что перекрытие – это количество общих  $n$ -грамм в  $C$  и  $R$  (суммирование ведется по различным  $n$ -граммам).

Штраф за краткость имеет вид:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & c \leq r \end{cases} \quad (5)$$

Здесь  $r$  это общая длина корпуса истинных предложений, а  $c$  это общая длина корпуса кандидатов. Однако в данной работе мы будем рассматривать метрику BLEU только для пары  $(C, R)$  и усреднять BLEU для пар по всему корпусу. Таким образом BLEU считается как произведение BP и геометрического среднего точностей  $p_n$  с весами  $\omega_n$

$$\text{BLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^N w_n \log p_n \right) \quad (6)$$

Обычно  $w_n = 1/N, N = 4$ .

### 3.1.1 Пример расчета метрики BLEU

Проиллюстрируем приведенные обозначения на примере:

$R = \text{"Петя взял яблоко и яблоко оказалось вкусным"}$

$C = \text{"Яблоко"}$

Получаем, что

$$O_1 = \min(\text{Count}_C(\text{яблоко}), \text{Count}_R(\text{яблоко})) = \min(1, 2) = 1$$

$$p_1 = \frac{1}{1} = 1, \text{BP} = e^{-6}$$

тогда если положить  $N = 1, \omega_1 = 1$  получаем

$$\text{BLUE} = e^{-6} \approx 0.002$$

Этот пример показывает, что несмотря на то, что точность предсказания максимально высокая, оно оказывается плохим с точки зрения метрики BLEU из-за того, что слишком короткое. Можно сказать, что в данной метрике штраф за краткость играет роль полноты.

## 4 Предлагаемый метод

### 4.1 BLEU в матричном виде

Для вывода нижней границы необходимо переписать формулу выше в матричной форме. Рассмотрим сгенерированный текст Candidate и истинный текст Reference и положим размер словаря равным  $v$ . Обозначим за  $x$  матрицу размера  $[\text{len}_C \times v]$ , которая содержит one-hot вектора для каждого слова в кандидате  $C$ . Аналогично обозначим за  $y$  матрицу размера  $[\text{len}_R \times v]$ , которая содержит one-hot вектора для каждого слова в истинном тексте. Где  $\text{len}_C$  и  $\text{len}_R$  - длины Candidate и Reference текста соответственно.

Для каждой 1-граммы мы считаем матрицы  $S^1 = xx^T$  и  $P^1 = yx^T$  размеров  $[\text{len}_C \times \text{len}_C]$  и  $[\text{len}_R \times \text{len}_C]$ , соответственно:

$$S_{i,j}^1 = x_i x_j \quad P_{i,j}^1 = y_i x_j$$

Аналогично определим матрицы для всех n-грамм.

$$S_{i,j}^n = \prod_{k=0}^{n-1} x_{i+k} x_{j+k} \quad P_{i,j}^n = \prod_{k=0}^{n-1} y_{i+k} x_{j+k} \quad (7)$$

Соответствующие размеры равны  $[\text{len}_C - n + 1 \times \text{len}_C - n + 1]$  и  $[\text{len}_R - n + 1 \times \text{len}_C - n + 1]$ . Важно отметить, что элементы этих матриц обладают следующим свойством:

$$S_{i,j}^n = \begin{cases} 1 & \text{if } Cand_i^n = Cand_j^n \\ 0 & \text{иначе} \end{cases} \quad (8)$$

$$P_{i,j}^n = \begin{cases} 1 & \text{if } Ref_i^n = Cand_j^n \\ 0 & \text{иначе} \end{cases} \quad (9)$$

Здесь  $Cand_i^n(Ref_i^n)$  это n-грамма на позиции  $i$  в предложении  $C(R)$ . Для каждой n-граммы для каждой позиции в  $C$  мы считаем число вхождений этой n-граммы в  $C$  или  $R$ .

$$v_i^{x,n} = \sum_{j=0}^{\text{len}_C-1} S_{j,i}^n \quad v_i^{y,n} = \sum_{j=0}^{\text{len}_y-1} P_{j,i}^n \quad (10)$$



Наконец мы используем полученные формулы для расчета перекрытий

$$O_n = \sum_{i=1}^{\text{len}_x} \frac{\min(v_i^{x,n}, v_i^{y,n})}{v_i^{x,n}} = \quad (11)$$

$$\sum_{i=1}^{\text{len}_x} \min(1, \frac{v_i^{y,n}}{v_i^{x,n}}) = \sum_{i=1}^{\text{len}_x} \min\left(1, \frac{\sum_{j=0}^{\text{len}_y-1} \prod_{k=0}^{n-1} y_{i+k} x_{j+k}}{\sum_{j=0}^{\text{len}_C-1} \prod_{k=0}^{n-1} x_{i+k} x_{j+k}}\right) \quad (12)$$

Доказательство того, что (4) и (11) равны, приведено в приложении В. Теперь мы можем посчитать модифицированные точности:

$$p_n = \frac{O_n}{\text{len}_x - n + 1} \quad (13)$$

Таким образом формула для расчета метрики BLEU переписана в матричном виде и можно продолжить вывод для BLB.

Пример расчета метрики BLEU в матричном виде приведен в приложении А

## 4.2 Нижняя граница математического ожидания метрики BLEU

Проведем вывод BLB в следующих упрощающих предположениях:

1. Матрица  $p_x$  размера  $[\text{len}_C \times \text{vocab\_size}]$ , которая описывает распределение на словах в последовательности  $C$ , детерминистически зависит от входной последовательности (то есть, например, отсутствует стохастичность во время декодирования).
2. Все слова в истинной последовательности  $R$  уникальны и кодируются one-hot векторами с помощью матрицы  $y$  размера  $[\text{len}_R \times \text{vocab\_size}]$

Если мы рассмотрим стандартную seq2seq архитектуру [4], то первое предположение значит, что как только мы закодировали контекст декодер сразу детерминистически выдает матрицу вероятностей  $p_x$ . Seq2seq модели во время тренировки обычно используют teacher-forcing [7] и в этом случае первое предположение сохраняется. Когда мы используем teacher forcing мы также игнорируем “штраф за краткость” (обозначается как BP), поэтому в дальнейшем выводе мы его опускаем (положим его равным 1).

Стоит отметить, что существуют более изощренные методы для тренировки seq2seq моделей, такие как например scheduled sampling [13]. В этом случае вычислительный граф декодера включает в себя стохастические вершины [12] и генерация матрицы  $p_x$  по входному тексту является случайным процессом, поэтому первое предположение не сохраняется.

Обозначим за  $p_y$  матрицу состоящую из one-hot векторов которые представляют последовательность  $R$ . Приведем план вывода BLB:

1. Записать математическое ожидание метрики BLEU используя распределение  $p_x$  и вырожденное распределение  $p_y$
2. Обозначить предположения необходимые для вывода нижней оценки для интеграла.
3. Рассмотреть случай вырожденного распределения  $p_x$  (матрица состоит one-hot векторов) и показать связь выведенной оценки с обычным BLEU.
4. Будем рассматривать значения  $p_x$  как параметры функции потерь (-BLB в данном случае). Градиенты функции потерь по параметрам могут быть посчитаны с помощью программ автоматического дифференцирования [14].

Для начала напишем формулу для математического ожидания  $\text{BLEU}(x, y)$  по распределению  $p_x$  при фиксированном  $y$ . Оно по определению равно третьему математическому ожиданию в формуле 2. Обозначим его как BLEU:

$$\begin{aligned} \text{BLEU} &= \mathbb{E}_{\text{output}} \text{BLEU}(x, y) = \mathbb{E}_{x \sim p_x} \text{BLEU}(x, y) = \\ &\mathbb{E}_{x \sim p_x} \left[ \exp \left( \sum_{n=1}^N \omega_n \log p_n \right) \right] \end{aligned} \quad (14)$$

BLEU это математическое ожидание произведения отдельных множителей соответствующих  $n$ -граммам. Оно может быть ограничено снизу произведением математических ожиданий.

$$\begin{aligned} \text{BLEU} &= \mathbb{E}_{x \sim p_x} \left[ \exp \left( \sum_{n=1}^N w_n \log p_n \right) \right] \\ &\geq \prod_{n=1}^N (\mathbb{E}_{x \sim p_x} [p_n])^{\omega_n} - 1 \end{aligned} \quad (15)$$

Последняя оценка тривиальна, т.к.  $p_n$  неотрицательны (и приведена лишь для строгости выкладок, однако на практике оценка оказывается довольно близкой к реальному значению). Рассмотрим один из множителей этого произведения. Из равенства (13)  $p_n$  с точностью до константы равен сумме перекрытий по  $n$ -граммам. Таким образом, для униграмм (BLEU1) имеем:

$$\mathbb{E}_{x \sim p_x} \left[ \sum_{i=1}^{\text{len}_x} O_1^i \right] = \mathbb{E}_{x \sim p_x} \left[ \sum_{i=1}^{\text{len}_x} \frac{\min(v^{y,1}, v^{x,1})}{v^{x,1}} \right] \quad (16)$$

Используя неравенство Йенсена и определения  $v^x, v^y$  мы можем вывести BLB для  $i$ -той компоненты перекрытия (приложение C):

$$\mathbb{E}_{x \sim p_x} [O_1^i] \geq \sum_n p_x^{in} \cdot \min \left( 1, \frac{\sum_j y_{jn}}{1 + \sum_{k \neq i} p_x^{kn}} \right) \quad (17)$$

Таким образом нижняя оценка для компонент  $O_1$  может быть вычислена используя  $p_x$  и  $y$ . Нижняя оценка для  $n$ -грамм для  $n > 1$  имеет похожий вид, что и для униграмм (приложение D):

$$\mathbb{E}_{x \sim p_x} [O_n^i] \geq \sum_{m_0} p_x^{i, m_0} \cdot \sum_{m_{n-1}} p_x^{i+n-1, m_{n-1}} \cdot \min \left( 1, \frac{\sum_j \prod_{k=0}^{n-1} y_{j+k, m_k}}{1 + \sum_{l \neq i} \prod_{k=0}^{n-1} p_x^{l+k, m_k}} \right) \quad (18)$$

Выведенная оценка обладает важным свойством, что если  $p_x$  и  $p_y$  вырожденные распределения (вероятности слов либо 1 либо 0), тогда оценка совпадает с точным значением BLEU( $x$ ,  $y$ ). Таким образом если мы хотим оптимизировать только один тип  $n$ -грамм (например, только 4-граммы), а не общий BLEU, мы можем использовать равенство 18. Если мы оптимизируем обычный BLEU до порядка  $n$  одновременно, мы можем использовать произведение 15 этих оценок, но для такой формулы не выполняется свойство того, что она совпадает с BLEU( $x$ ,  $y$ ) для вырожденных распределений.

## 5 Эксперименты и результаты

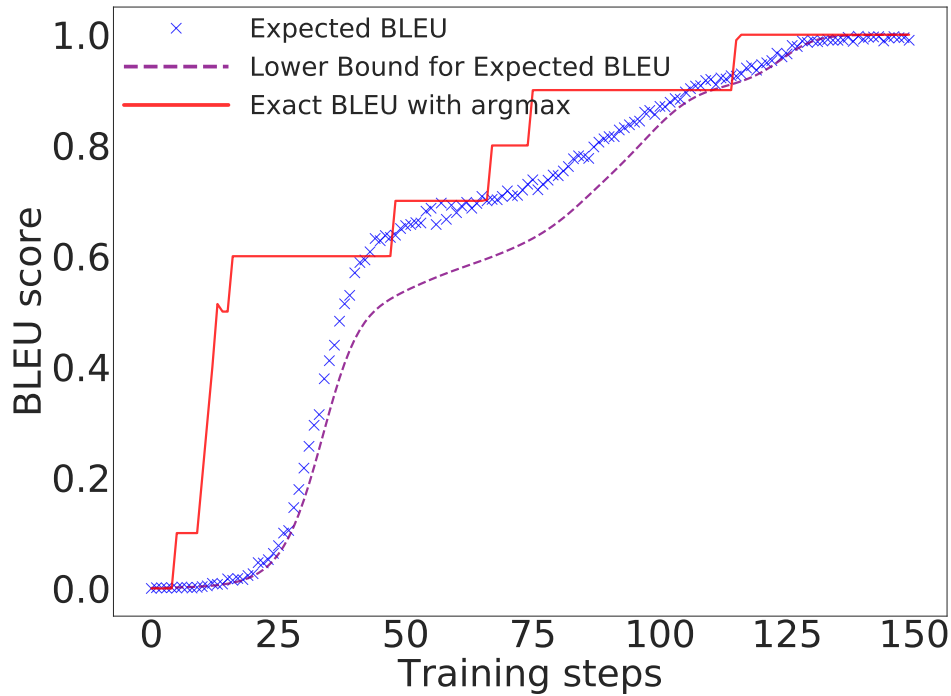
### 5.1 Модельная задача

Для того чтобы протестировать предложенный метод был поставлен следующий эксперимент.

Генерируется матрица распределения на словах  $p_x$  для перевода  $C$  и матрица one-hot векторов истинного текста  $r$ . Обе последовательности имеют длину 10. Размер словаря выбран равным 10000.

- Элементы матрицы  $p_x$  сгенерированы сэмпированием значений из  $\mathcal{N}(0, 1)$  а затем к ним построчно применяется функция *softmax*, чтобы получить распределение вероятностей на словах.
- Индексы истинного текста  $r$  сгенерированы сэмпированием с возвращением из равномерного распределения на множестве слов словаря.

В такой постановке нашей задачей является максимизация математического ожидания метрики BLEU между текстом сэмпированным из  $p_x$  и

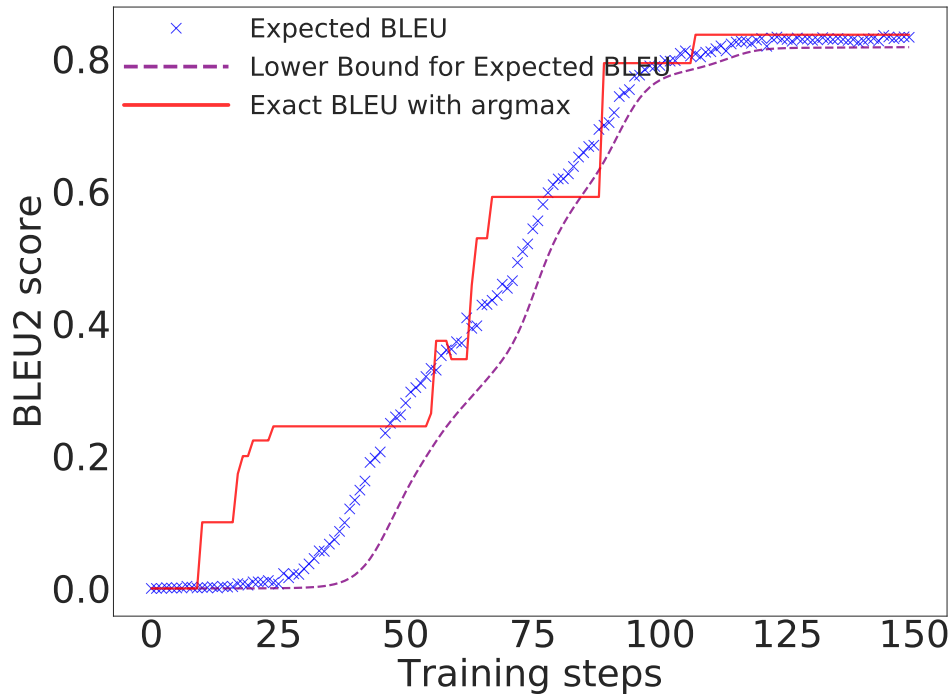


**Рис. 7:** Пример кривой обучения для униграмм – BLEU1 метрики. Красная линия – это точный BLEU полученный применением операции *argmax* к матрице  $p_x$  построчно с последующим вычислением BLEU. Фиолетовая пунктирная линия – это нижняя граница математического ожидания метрики BLEU (BLB). Линия из крестов обозначает математическое ожидание BLEU, полученное усреднением сэмплов из распределения  $p_x$ .

текстом полученным из  $r$ . Оптимизация ведется по параметра  $p_x$ . Для оптимизации используется нижняя граница математического ожидания метрики BLEU. Кривая обучения представлена на рисунке 7. График показывает скоррелированное поведение нижней оценки, точного значения метрики BLEU и его математического ожидания.

## 5.2 Результаты в задаче перевода

Мы также протестировали данный подход на датасетах для обучения машинного перевода немецко-английских пар предложений IWSLT'14 [15] и англо-немецких пар WMT'14 [16]. Для IWSLT'14 использовалась та же предобработка текста что и в работе [17]. Для WMT'14 использовалась предобработка предоставленная в датасете.



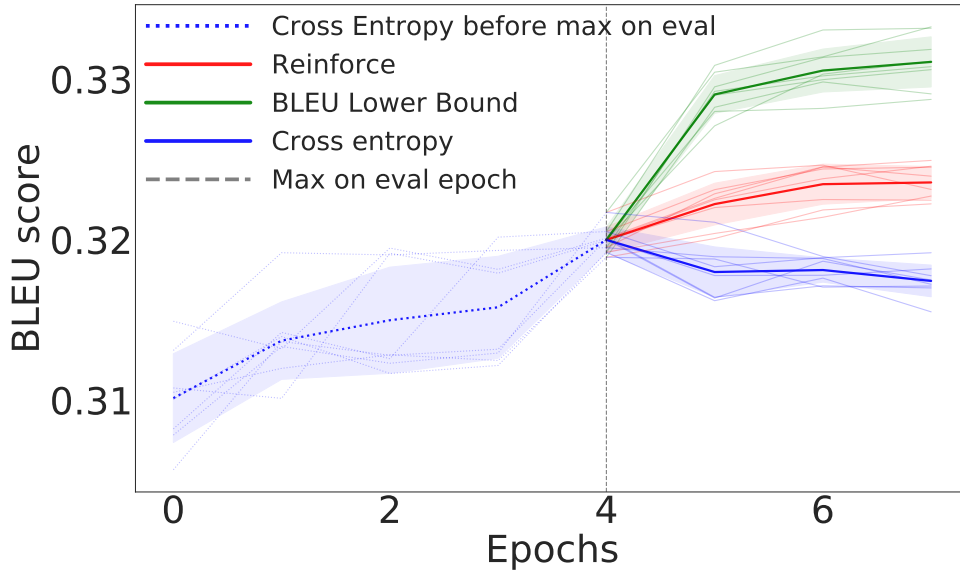
**Рис. 8:** Пример кривой обучения для биграмм (BLEU2). Линия из крестов показывает  $LB + 1$ .

	CE	RF	LB
1	$26.60 \pm 0.14$	$27.16 \pm 0.18$	<b><math>27.83 \pm 0.17</math></b>
5	$27.56 \pm 0.17$	$27.85 \pm 0.11$	<b><math>28.10 \pm 0.17</math></b>

**Таблица 1:** Результат на задаче перевода для датасета IWSLT’14 для трех способов оптимизации. Перекрестная энтропия (CE), REINFORCE (RF), BLEU lower bound (LB) (предлагаемый подход). Размер beamsize в лучевом поиске был равен 1 и 5. Результаты указаны по метрике BLEU на тестовом датасете.

	CE	LB
1	$17.12 \pm 0.2$	<b><math>18.65 \pm 0.13</math></b>
5	<b><math>19.23 \pm 0.19</math></b>	$19.03 \pm 0.15$

**Таблица 2:** Результат на задаче перевода для датасета WMT’14 для двух способов оптимизации. Перекрестная энтропия (CE), REINFORCE (RF), BLEU lower bound (LB) (предлагаемый подход). Размер beamsize в лучевом поиске был равен 1 и 5. Результаты указаны по метрике BLEU на тестовом датасете.



**Рис. 9:** Процесс обучения для разных способов оптимизации на датасете IWSLT'14 DE→EN [15]. Сначала модель оптимизировалась с помощью функции потерь перекрестной энтропии. Далее, начиная с лучших весов (качество измерялось на валидационном датасете), модель оптимизировалась с помощью перекрестной энтропии, BLB и с помощью REINFORCE. На графике указаны результаты по метрике BLEU на валидационном датасете.

### 5.2.1 Используемая модель

Была использована стандартная seq2seq модель [4]. Кодировщик представляет собой двунаправленный LSTM [18] с 256 нейронами в скрытом состоянии для датасета IWSLT'14 и с 512 нейронами в скрытом состоянии для датасета WMT'14. Декодер представляет собой LSTM с 256 нейронами в скрытом состоянии. В декодере использовался multiplicative attention [19], для вычисления контекстного вектора (взвешенная сумма скрытых состояний энкодера). Входом декодера на шаге  $t$  является конкатенация представления слова на шаге  $t - 1$  и нелинейного преобразования контекстного вектора и скрытого состояния с предыдущего шага.<sup>1</sup> Во время валидации использовалось жадное декодирование и лучевой поиск с beamsize равным 1 и 5.

<sup>1</sup>Имплементация доступна по ссылке [https://github.com/deepmipt/bleu\\_lower\\_bound](https://github.com/deepmipt/bleu_lower_bound)

### 5.2.2 Детали обучения модели

Для каждого эксперимента было сделано 8 запусков, чтобы рассчитать стандартное отклонение целевой метрики. Для оптимизации мы использовали метод оптимизации Adam [20] с learning rate равным  $10^{-4}$  для функций потерь перекрестной энтропии и BLB и  $10^{-5}$  для REINFORCE.

Процесс обучения модели состоит из 4 шагов:

1. Оптимизация модели с помощью перекрестной энтропии в течении 20 эпох
2. Выбор лучших весов модели с помощью валидационного датасета.
3. Дальнейшая оптимизация модели (начиная с найденных весов) в течении 5 эпох с помощью одного из трех подходов: перекрестная энтропия, REINFORCE, BLB
4. Выбор эпохи на которой модель показала лучшее качество по метрике BLEU на валидационном датасете.

Также важно отметить, что аддитивное сглаживание (additive smoothing) для точностей в расчете метрики BLEU повышает стабильность обучения на задаче машинного перевода.

$$\text{LB} \left[ \frac{O_n}{\text{len}_x - n + 1} \right] \rightarrow \frac{\text{LB}[O_n] + 1}{\text{len}_x - n + 2}$$

По своей сути это стандартный трюк добавления константы под логарифм (геометрическое среднее  $p_n$  вычисляется как экспонента от суммы логарифмов), однако в экспериментах вычислительная нестабильность проявлялась только при  $n > 1$ .

Результаты на тестовой выборке датасета IWSLT'14 приведены в таблице 1 в формате “mean  $\pm$  std”. Оптимизация с помощью BLB приводит к более высоким значения метрики BLEU на тестовой выборке для обоих размеров beamsize.

Результаты на тестовой выборке датасета WMT'14 (EN-DE) приведены в таблице 2 в формате “mean  $\pm$  std”. Оптимизация с помощью BLB



приводит к более высоким значениям метрики BLEU на тестовой выборке, если значение beamsize в лучевом поиске равно 1 и не приводит для beamsize равного 5.

## 6 Выводы

Предложена вычислительно эффективная, дифференцируемая нижняя оценка для математического ожидания метрики BLEU. На задаче перевода показано, что оптимизация такой оценки ведет к повышению значения метрики BLEU в сравнении с альтернативами (cross-entropy, REINFORCE). Для того, чтобы использовать BLB необходимы совсем незначительные изменения процесса обучения перекрестной энтропией: на последних эпохах заменить функцию потерь перекрестную энтропию на BLB. Имплементация алгоритма доступна в открытом доступе по [ссылке](#).

## Список литературы

- [1] Bleu: a method for automatic evaluation of machine translation / Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. — 2002.
- [2] Lin, C.-Y. Rouge: A package for automatic evaluation of summaries. — 2004.
- [3] Liu, C.-W. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. — 2016.
- [4] Sutskever, I. Sequence to sequence learning with neural networks. — 2014.
- [5] Barone, A. V. M. Deep architectures for neural machine translation. — 2017.
- [6] Rush, A. M. A neural attention model for abstractive sentence summarization. — 2015.
- [7] Williams, R. J. A learning algorithm for continually running fully recurrent neural networks / Ronald J. Williams, David Zipser. — 1989.
- [8] Sequence level training with recurrent neural networks / Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, Wojciech Zaremba // CoRR. — 2015. — Vol. abs/1511.06732. — <http://arxiv.org/abs/1511.06732>.
- [9] Wiseman, S. Sequence-to-sequence learning as beam-search optimization / Sam Wiseman, Alexander M. Rush // CoRR. — 2016. — Vol. abs/1606.02960. — <http://arxiv.org/abs/1606.02960>.
- [10] Minimum risk training for neural machine translation / Shiqi Shen, Yong Cheng, Zhongjun He et al. // CoRR. — 2015. — Vol. abs/1512.02433. — <http://arxiv.org/abs/1512.02433>.
- [11] Schulman, J. Gradient estimation using stochastic computation graphs. — 2015.

- [12] Golikov, E. Using stochastic computation graphs formalism for optimization of sequence-to-sequence model. — 2017.
- [13] Bengio, S. Scheduled sampling for sequence prediction with recurrent neural networks. — 2015.
- [14] Automatic differentiation in pytorch / Adam Paszke, Sam Gross, Soumith Chintala et al. — 2017.
- [15] Report on the 11th iwslt evaluation campaign, iwslt 201 / Mauro Cettolo, Jan Niehues, Sebastian Stuker et al. — 2014.
- [16] ACL2014. Ninth workshop on statistical machine translation. — <http://www.statmt.org/wmt14/index.html>.
- [17] Wiseman, S. Sequence-to-sequence learning as beam-search optimization. — 2016.
- [18] Hochreiter, S. Long short-term memory / Sepp Hochreiter, Jürgen Schmidhuber // Neural Computation. — 1997. — P. 9:1735–1780.
- [19] Luong, M.-T. Effective approaches to attention-based neural machine translation. — 2015.
- [20] Kingma, D. P. Adam: A method for stochastic optimization. — 2014.

## А Пример расчета метрики BLEU в матричном виде

$$C = \text{"}abba\text{"} \quad R = \text{"}abb\text{"}$$

$$x = \begin{matrix} & a & b \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \end{matrix} \quad y = \begin{matrix} & a & b \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \end{matrix}$$

Для униграмм имеем:

$$S^1 = \begin{matrix} & a & b & b & a \\ \begin{matrix} a \\ b \\ b \\ a \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad P^1 = \begin{matrix} & a & b & b & a \\ \begin{matrix} a \\ b \\ b \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$v^{x,1} = \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \quad v^{y,1} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

Для биграмм:

$$S^2 = \begin{matrix} & ab & bb & ba \\ \begin{matrix} ab \\ bb \\ ba \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad P^2 = \begin{matrix} & ab & bb & ba \\ \begin{matrix} ab \\ bb \end{matrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{matrix}$$

$$v^{x,2} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad v^{y,2} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Для триграмм:

$$S^3 = \begin{matrix} & abb & bba \\ \begin{matrix} abb \\ bba \end{matrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{matrix} \quad P^3 = \begin{matrix} & abb & bba \\ \begin{matrix} abb \end{matrix} & \begin{pmatrix} 1 & 0 \end{pmatrix} \end{matrix}$$

$$v^{x,3} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad v^{y,3} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

## В Эквивалентность формул для перекрытий $O_n$

Пусть  $W$  обозначает множество  $n$ -грамм для  $C$ .  $I_\omega$  обозначает множество начальных индексов  $n$ -грамм  $\omega$  в  $C$ :

$$O_n = \sum_{\omega \in W} \sum_{i \in I_\omega} \frac{\min(v_i^{y,n}, v_i^{x,n})}{v_i^{x,n}} = (*)$$

Рассмотрим  $n$ -грамму  $\omega$  и  $i$  - индекс где  $n$ -грамма  $\omega$  начинается в переведенном предложении. Вспомним, что  $v_i^{x,n}$  (и  $v_i^{y,n}$ ) обозначают число вхождений  $n$ -граммы  $\omega$  в предложение перевода. Рассмотрим  $i, j \in I_\omega$ , тогда по определению

$$\begin{aligned} v_i^{x,n} &= v_j^{x,n} = \text{Count}_C(\omega) = |I_\omega| \\ v_i^{y,n} &= v_j^{y,n} = \text{Count}_R(\omega) \\ (*) &= \sum_{\omega \in W} \sum_{i \in I_\omega} \frac{\min(\text{Count}_R(\omega), \text{Count}_C(\omega))}{\text{Count}_C(\omega)} = \\ &= \sum_{\omega \in W} \min(\text{Count}_R(\omega), \text{Count}_C(\omega)) \end{aligned}$$

### В.1 Пример

Рассмотрим гипотезу

$$C = \text{“Mike took an apple and the apple was delicious”}$$

Проведем пример для случая униграмм  $n = 1$ . Рассмотрим слово  $\omega = \text{apple}$ , тогда  $I_\omega = \{3, 6\}$  - начальные индексы слова apple в данном предложении. Количество вхождений слова  $\omega$  в предложение равно

$$\begin{aligned} 2 &= |I_\omega| = \text{Count}_C(w) = \\ &= v_3^{x,1} = v_6^{x,1} \end{aligned}$$

по определению  $v$ .

## С Нижняя граница для униграмм

Вывод нижней границы для униграммной метрики BLEU:

$$\mathbb{E}_{x \sim p_x, y \sim p_y} O_1^i = \mathbb{E}_{x \sim p_x, y \sim p_y} \left[ \min \left( 1, \frac{\sum_j x_i y_j}{\sum_k x_i x_k} \right) \right] \quad (19)$$

Разбиваем математическое ожидание на два интеграла, один из них по случайной переменной  $x_i$  а оставшийся по всем остальным. Распределение на  $y$  опущено здесь и далее:

$$\begin{aligned} &= \mathbb{E}_{x_{k,k \neq i}} \left[ \sum_n p_x^{in} \min \left( 1, \frac{\sum_j y_{jn}}{\sum_k x_i x_k} \right) \right] \\ &= \sum_n p_x^{in} \cdot \mathbb{E}_{x_{k,k \neq i}} \left[ \min \left( 1, \frac{\sum_j y_{jn}}{1 + \sum_{k \neq i} x_i x_k} \right) \right] \end{aligned} \quad (20)$$

Выражение в квадратных скобках может рассматриваться как функция от переменной  $x_k$  на  $[0, 1]^n$ . Эта функция является вогнутой (приложение E), поэтому мы можем применить неравенство Йенсена чтобы вывести нижнюю границу:

$$\begin{aligned} &= \sum_n p_x^{in} \cdot \mathbb{E}_{x_{k,k \neq i}} \left[ \min \left( 1, \frac{\sum_j y_{jn}}{1 + \sum_{k \neq i} x_i x_k} \right) \right] \\ &\geq \sum_n p_x^{in} \cdot \min \left( 1, \frac{\sum_j y_{jn}}{1 + \sum_{k \neq i} p_x^{kn}} \right) \end{aligned} \quad (21)$$

## D Обобщение для n-грамм

Запишем явно математическое ожидание по случайным величинам  $\{x_{i+p} : p \in \{0..n-1\}\}$  для  $i$ -ой компоненты перекрытия  $O_n$

$$\mathcal{L}_i = x_{k, k \notin \{i+p: p \in \{0..n-1\}\}}$$

$$\begin{aligned} &\mathbb{E}_{x \sim p_x} [O_n^i] = \\ &\mathbb{E}_{\mathcal{L}_i} \left[ \sum_{m_0} p_x^{i, m_0} \dots \sum_{m_{n-1}} p_x^{i+n-1, m_{n-1}} O_n^i \right] \end{aligned} \quad (22)$$

Пусть  $s$  это некоторый индекс и  $s \notin \{i + p : p \in \{0..n - 1\}\}$ : Рассмотрим два множества  $A_{s,i} = \{l : l \neq i, s \in [l..l + n - 1]\}$  и  $B_{s,i} = \{l : l \neq i, s \notin [l..l + n - 1]\}$  и перепишем

$$O_n^i = \min\left(1, \frac{\sum_j \prod_{k=0}^{n-1} y_{j+k, m_k}}{1 + \sum_{A_{s,i}} \prod_{k=0}^{n-1} x_{i+k} x_{l+k} + \sum_{B_{s,i}} \prod_{k=0}^{n-1} x_{i+k} x_{l+k}}\right)$$

Поменяем порядок интегрирования и запишем отдельно математическое ожидание по  $p_x^s$ ,

$$\sum_{m_0} p_x^{i, m_0} \dots \sum_{m_{n-1}} p_x^{i+n-1, m_n} \mathbb{E}_{x_{k, k \notin \{i+p: p \in \{0..n-1\}\}, k \neq s}} \mathbb{E}_{p_x^s} O_n^i \quad (23)$$

Математическое ожидание по случайной величине  $x^s$  имеет следующий вид:

$$\min\left(1, \frac{A}{1 + B + \vec{c}x_s}\right)$$

Где  $A, B$  некоторые константы и  $\vec{c}$  константный вектор:

$$\frac{A}{1 + B + \vec{c}x_s}$$

Поскольку это выпуклая функция (приложение [Е](#)), мы можем применить неравенство Йенсена последовательно для всех  $s \notin \{i + p : p \in \{0..n - 1\}\}$ .

Новыми  $O_n^i$  будут

$$O_i'^n = \min\left(1, \frac{\sum_j \prod_{k=0}^{n-1} y_{j+k, m_k}}{1 + \sum_{l \neq i} \prod_{k=0}^{n-1} p_x^{l+k, m_k}}\right)$$

Итоговое неравенство:

$$\mathbb{E}_{x \sim p_x} [O_n^i] \geq \sum_{m_0} p_x^{i, m_0} \dots \sum_{m_{n-1}} p_x^{i+n-1, m_n} O_i'^n$$

## Е Доказательство выпуклости

Обозначим за  $a$  фиксированный вектор, а  $A, B$  некоторые положительные константы:

$$\frac{A}{B + a((1 - \alpha)x_s + \alpha y_s)} \leq \frac{(1 - \alpha)A}{B + ax_s} + \frac{\alpha A}{B + ay_s} \quad (24)$$

Без ограничения общности:

$$\frac{1}{1 + a((1 - \alpha)x_s + \alpha y_s)} \leq \frac{1 - \alpha}{1 + ax_s} + \frac{\alpha}{1 + ay_s} \quad (25)$$

Обозначим  $S = ax_s$ ,  $T = ay_s$ :

$$\begin{aligned} \frac{1}{1 + (1 - \alpha)S + \alpha T} &\leq \frac{1 - \alpha}{1 + S} + \frac{\alpha}{1 + T} \\ \frac{1}{1 + (1 - \alpha)S + \alpha T} &\leq \frac{(1 - \alpha)(1 + T) + \alpha(1 + S)}{(1 + T)(1 + S)} \\ \frac{1}{1 + (1 - \alpha)S + \alpha T} &\leq \frac{1 + (1 - \alpha)T + \alpha S}{(1 + T)(1 + S)} \\ \ln(1 + T) + \ln(1 + S) &\leq \\ \ln(1 + (1 - \alpha)S + \alpha T) + \ln(1 + (1 - \alpha)T + \alpha S) \end{aligned}$$

Поскольку функция  $\ln(1 + x)$  выпуклая:

$$\begin{aligned} \ln(1 + (1 - \alpha)S + \alpha T) &\geq \\ (1 - \alpha) \ln(1 + S) + \alpha \ln(1 + T) & \\ \ln(1 + (1 - \alpha)T + \alpha S) &\geq (1 - \alpha) \ln(1 + T) + \alpha \ln(1 + S) \end{aligned}$$

## Ф Полные результаты экспериментов на датасете IWSLT



LB	CE
28,16	26,67
27,73	26,73
27,65	26,42
28,08	26,73
27,88	26,70
27,95	26,36
27,7	26,64
27,85	26,57

LB	CE
27,87 $\pm$ 0,18	26,60 $\pm$ 0,14

**Таблица 3:** Результаты по метрике BLEU для двух способов оптимизации: Перекрестная энтропия (CE), BLEU lower bound (LB). В таблице слева результаты приведены для восьми экспериментов с разной начальной инициализацией (для различных значений random seed). В таблице справа указаны агрегированные результаты в формате “mean  $\pm$  std”.

Значения получены для лучевого поиска с beamsize=1 (жадное декодирование).

LB	CE
28,33	27,72
28,05	27,77
27,84	27,27
28,32	27,63
28,11	27,65
28,18	27,52
27,93	27,51
28,06	27,41

LB	CE
28,10 $\pm$ 0,17	27,56 $\pm$ 0,16

**Таблица 4:** Результаты по метрике BLEU для двух способов оптимизации: Перекрестная энтропия (CE), BLEU lower bound (LB). В таблице слева результаты приведены для восьми экспериментов с разной начальной инициализацией (для различных значений random seed). В таблице справа указаны агрегированные результаты в формате “mean  $\pm$  std”.

Значения получены для лучевого поиска с beamsize=5.

RF	CE
27,42	26,67
27,09	26,73
27,03	26,42
27,15	26,73
27,29	26,70
26,82	26,36
27,19	26,64
27,27	26,57

RF	CE
27,16 $\pm$ 0,18	26,60 $\pm$ 0,14

**Таблица 5:** Результаты по метрике BLEU для двух способов оптимизации: Перекрестная энтропия (CE), правило REINFORCE (RF). В таблице слева результаты приведены для восьми экспериментов с разной начальной инициализацией (для различных значений random seed). В таблице справа указаны агрегированные результаты в формате “mean  $\pm$  std”.

Значения получены для лучевого поиска с beamsize=1 (жадное декодирование).

RF	CE
28,06	27,72
27,9	27,77
27,66	27,27
27,86	27,63
27,83	27,65
27,8	27,52
27,83	27,51
27,89	27,41

RF	CE
27,85 $\pm$ 0,11	27,56 $\pm$ 0,16

**Таблица 6:** Результаты по метрике BLEU для двух способов оптимизации: Перекрестная энтропия (CE), правило REINFORCE (RF). В таблице слева результаты приведены для восьми экспериментов с разной начальной инициализацией (для различных значений random seed). В таблице справа указаны агрегированные результаты в формате “mean  $\pm$  std”.

Значения получены для лучевого поиска с beamsize=5.