© 2022 г. Н.А. СКАЧКОВ (nikolaj-skachkov@yandex.ru)
К.В. ВОРОНЦОВ, д-р физ-мат. наук (vokov@forecsys.ru)
(Федеральный исследовательский центр
“Информатика и управление” РАН, Москва)

УЛУЧШЕНИЕ КАЧЕСТВА МАШИННОГО ПЕРЕВОДА С ИСПОЛЬЗОВАНИЕМ ОБРАТНОЙ МОДЕЛИ

Машинный перевод — это задача обработки текстов естественного языка, ставящая своей целью перевод входного текста с одного языка на другой язык в автоматическом режиме. Известные на данный момент модели машинного перевода показывают достаточно высокое качество перевода между крупными языками, однако для более мелких языковых направлений, представленных меньшим количеством данных, задача все еще не решена. Для борьбы с различными ошибками автоматических систем перевода применяются разные методы. В данной работе рассматриваются подходы, использующие переводные модели обратных языковых направлений и улучшающие согласованность между переводами одного текста с помощью прямых и обратных им моделей перевода. В работе представлено общее теоретическое обоснование для таких методов с точки зрения решения задачи максимизации правдоподобия, а также предложен способ стабильного обучения современных моделей с использованием циклических переводов.

Ключевые слова: машинный перевод, нейронная сеть, стохастический градиентный спуск, вероятностное моделирование, максимум правдоподобия, выбор по значимости, циклические переводы, дообучение модели

DOI: 10.31857/S0005231022120042, **EDN:** KRVZJE

1. Вступление

Задача машинного перевода является важной и сложной задачей анализа и генерации текстов естественного языка. Ручной перевод текстов является дорогостоящим и трудоемким процессом, так как требует привлечения специалистов с глубоким знанием нескольких языков.

В основе современных моделей автоматического перевода лежат нейросетевые модели, обученные на большом количестве пар: предложение и его перевод [1]. На основе большого объема данных модели перевода выучивают межъязыковые связи и языковые модели, которые необходимы для генерации качественного перевода. При этом в самом процессе обучения переводная модель учится пословно воспроизводить поданные ей переводы.

Одним из важных аспектов при оценке качества автоматического перевода является согласованность или смысловая схожесть текстов, полученных с помощью перевода между разными языками одного и того же текста. Например, если при переводе какого-то текста с английского на русский и обратно

на английский получить результат, значительно отличающийся от исходного текста, это свидетельствует о низком качестве моделей перевода между данными языками.

Для избежания описанных проблем в переводах существует метод, напрямую улучшающий согласованность переводов прямой и обратной моделей перевода [2]. Для этого предлагается учить модель генерировать такие переводы, чтобы перевод обратной модели больше совпадал с исходным текстом. При этом авторы обучают такую модель с помощью алгоритма REINFORCE с эмпирически выбранной функцией награды. Оказалось, что такой метод увеличивает не только согласованность, но и качество перевода в целом за счет предоставления прямой модели во время обучения информации, которую несет обратная модель.

Однако применение описанного метода сопряжено с некоторыми трудностями. Одна из них заключается в необходимости одновременно хранить градиенты для прямой и обратной моделей на каждой итерации обучения. В изначальном методе предполагается использовать рекуррентные модели перевода [3], однако для современных архитектур такие требования делают метод труднопримени́мым на практике, так как гиперпараметры прямой модели для наиболее эффективного обучения выбираются так, чтобы использовать всю возможную память вычислительных устройств [4]. Другая проблема описанного подхода заключается в том, что на первых итерациях обучения градиент для прямой модели обладает высокой дисперсией. По этой причине в [2] для стабилизации обучения оценивается значимость обратного перевода с помощью дополнительных языковых моделей, что требует еще больше вычислительных ресурсов и времени.

В данной работе представлен теоретически обоснованный метод обучения с использованием обратной модели, выведенный из правдоподобия циклических переводов. Полученная формула функции потерь отличается от формулы в [2]. Кроме того, в ней не используются эвристики для стабилизации обучения с помощью языковых моделей. Проблема высокой дисперсии градиентов, даваемых предложенной функцией потерь решается с помощью дообучения уже предобученной модели перевода. Данное решение экономит вычислительные ресурсы, так как нет необходимости хранить в памяти дополнительные параметры языковых моделей и статистики оптимизатора для обратной модели, а сам процесс дообучения сходится существенно быстрее, чем обучение с нуля. Все это позволяет обучать прямую модель совместно с обратной, используя современные архитектуры моделей перевода, не уменьшая их в размере.

Также в работе показано, каким образом обучение с обратной моделью сказывается на качестве машинного перевода, а также на согласованности переводов текстов между прямой и обратной ей моделями. Эксперименты проводятся на русско-казахском и англо-финском языковых направлениях.

2. Описание подхода

В данном разделе приведены вероятностные модели, лежащие в основе построения алгоритмов автоматического перевода, а также предлагается обобщение использования обратных моделей через сведение к задаче моделирования правдоподобия циклических переводов.

2.1. Задача машинного перевода

Опишем для начала общую постановку задачи машинного перевода. Пусть дано множество пар текстов на двух языках $\{(x_i, y_i)\}_{i=1}^{N}$. Обозначим, что тексты x_i принадлежат языку входа (ЯВ), а тексты y_i принадлежат целевому языку (ЦЯ). Для обучения модели перевода необходимо максимизировать правдоподобие переводных текстов, при условии входных текстов. Записывая данное требование в виде максимизации логарифма правдоподобия, получим:

$$\sum_{i=1}^{N} \log P_\theta(y_i|x_i) \longrightarrow \max_\theta,$$

где θ — параметры переводной модели.

Для модели перевода необходимо, чтобы модель не только могла давать оценку вероятности переводного текста, но и давала возможность получить сам перевод. Частым решением является использование авторегрессионных моделей [1]. В данном подходе генерация перевода происходит итеративно слева-направо. При этом оценка вероятности перевода с этим требованием записывается так:

$$\log P_\theta(y|x) = \sum_{t=1}^{|y|} \log P_\theta(y^t|y^{<t}, x), \tag{1}$$

где y^t — это t-е слово перевода, а $y^{<t}$ — это префикс перевода для t-го слова.

Авторегрессионность является достаточно сильным ограничением, однако благодаря итеративной генерации можно избежать полного перебора гипотез перевода в поисках гипотезы с максимальной вероятностью. Далее в данной работе все модели перевода будут предполагаться авторегрессионными.

2.2. Использование обратной модели

В данной работе как и в оригинальном методе будем рассматривать рост согласованности моделей как увеличения совпадения циклического перевода через целевой язык с оригинальным текстом. Для этого рассмотрим конструкцию порождения циклического перевода ЯВ→ ЦЯ→ ЯВ. В нем текст на языке входа прямой моделью переводится на целевой язык и, далее, переводится обратной моделью снова на язык входа. Для построения вероятностной модели нужно записать необходимость совпадения полученного циклического перевода с исходным текстом.

Для этого введем понятие вероятности циклического перевода следующим образом:

$$P_{\text{cycle}}(x'|x) = \mathbb{E}_{y\sim P_\Theta(y|x)} P'(x'|y), \tag{2}$$

где $P_\Theta(y|x)$ — параметризованная прямая модель перевода, а $P'(x|y)$ — обратная модель. Такая вероятность показывает то, какова в среднем по всем прямым переводам модели вероятность получения циклического перевода x' из текста x.

Несмотря на полезность введенного понятия для построения вероятностной модели, вычислить его невозможно, так как подсчет среднего по всем возможным переводам приводит к суммированию по бесконечному множеству текстов на целевом языке. Для получения несмещенной оценки этого выражения, которое может понадобиться для оценки функции потерь на тестовой выборке, нужно воспользоваться процедурой выбора по значимости.

Введя вероятность циклического перевода $P_{\text{cycle}}(x'|x)$, можно перейти к записи оптимизируемого функционала, отражающего необходимость совпадения циклического перевода с исходным текстом

$$\log P_{\text{cycle}}(x|x) = \log \mathbb{E}_{y\sim P_\Theta(y|x)} P'(x|y) \longrightarrow \max_\Theta .$$

Для обучения нейросетевых моделей методом градиентного спуска необходимо получить значение градиента данной функции потерь. При этом можно заметить, что после произведенного логарифмирования выражение более невозможно оценить несмещенно с помощью процедуры выбора по значимости. По этой причине перейдем к максимизации нижней оценки функции потерь, воспользовавшись при этом неравенством Йенсена для вогнутых функций. Функция логарифма является вогнутой, а вероятностное распределение образует выпуклую комбинацию, следовательно

$$\log \mathbb{E}_{y\sim P_\Theta(y|x)} P'(x|y) \geqslant \mathbb{E}_{y\sim P_\Theta(y|x)} \log P'(x|y) =: L(x).$$

Так как был осуществлен переход к нижней оценке, теперь необходимо получить значения градиентов именно для нее. Поэтому представим математическое ожидание в виде интеграла, и внесем градиент, являющийся линейным оператором, внутрь

$$\nabla_\Theta L(x) = \boldsymbol{\nabla_\Theta} \int P_\Theta(y|x) \log P'(x|y) dy = \int \log P'(x|y) \boldsymbol{\nabla_\Theta P_\Theta(y|x)} dy.$$

Чтобы избежать суммирования по бесконечному множеству, при подсчете градиента также нужно воспользоваться процедурой выбора по значимости. Для этого данное представление градиента нижней оценки функции потерь представим в виде математического ожидания по некоторому распределению, воспользовавшись следующим преобразованием:

$$\frac{\partial \log f(x)}{\partial x} = \frac{1}{f(x)} \frac{\partial f(x)}{\partial x}.$$

Применив данное преобразование, получим следующий вид градиента нижней оценки функции потерь:

$$\nabla_\Theta L(x) = \int \log P'(x|y)\boldsymbol{P_\Theta(y|x)\nabla_\Theta \log P_\Theta(y|x)}dy = \\ = \mathbb{E}_{y\sim P_\Theta(y|x)} \log P'(x|y)\nabla_\Theta \log P_\Theta(y|x).$$

Теперь есть возможность оценить данное выражение значением на одном примере с помощью процедуры выбора по значимости

$$\nabla_\Theta L(x) \approx \log P'(x|y)\nabla_\Theta \log P_\Theta(y|x), \quad y \sim P_\Theta(y|x). \tag{3}$$

Если рассмотреть $\mathcal{L}(x, y)$ — функцию потерь прямой модели при обучении без циклических переводов (1), то полученная оценка градиента (3) может быть представлена как градиент при обучении без циклических переводов, умноженный на некоторый коэффициент

$$\nabla_\Theta \mathcal{L}(x, y) = \nabla_\Theta \log P_\Theta(y|x)$$
$$\nabla_\Theta L(x) \approx w(x, y)\nabla_\Theta \log P_\Theta(y|x), \quad y \sim P_\Theta(y|x),\ w = \log P'(x|y).$$

Представление градиента функции потерь в виде градиента прямой модели, умноженной на коэффициент, позволяет дать интерпретацию полученным формулам: чем больше вероятность получения исходного текста при обратном переводе из y, тем более прямая модель будет стремиться переводить x как y в сравнении с обучением без циклической функции потерь (1).

При полученном визуальном сходстве градиентов функций потерь прямой модели перевода (1) и модели с циклическими переводами (2) в последней вместо реальных текстов используются сгенерированные с помощью прямой модели переводы.

Можно заметить, что представленное выражение градиента (3) имеет сходство с выражениями градиентов из оригинальной работы [2], при том что ее авторы получили градиенты для обучения, используя эвристики. В данной же работе предложена исходная вероятностная модель, из которой получены градиенты для оптимизации. Данная вероятностная модель обобщает использование обратной модели перевода и дает возможность для развития данного метода в будущем.

3. Используемые методы

3.1. Архитектура модели

В основе современных моделей перевода лежат нейронные сети архитектуры Transformer [5]. Архитектура Transformer, в свою очередь, состоит из кодировщика и декодировщика, каждый из которых состоит из последовательных применений блоков одинаковой структуры. Каждый блок обладает своим собственным набором параметров и применяется к каждому векторному представлению слова, полученного от предыдущего блока. В процессе работы кодировщика для каждого входного слова подается информация о

каждом слове из входного предложения. Эта информация помогает обогатить векторное представление слова с помощью контекстной информации. В процессе работы декодировщика подается информация только о словах префикса, предшествующего обрабатываемому слову. Это позволяет использовать декодировщик для авторегрессионной генерации (1).

Обучение модели заключается в минимизации функций потерь (1) и (2) методом Adam [6]. В основе данного метода лежит процедура стохастического градиентного спуска, при котором вычисляется градиент функции потерь по оптимизируемым параметрам и совершается спуск в направлении, противоположном вычисленному градиенту.

В данной работе для обучения модели перевода использовалась именно архитектура Transformer с оптимизатором Adam. Подробнее с особенностями поставленных экспериментов можно познакомиться в секции с экспериментами.

3.2. Оценка качества перевода

Для оценки качества переводов в экспериментах будет использоваться метрика BLEU [7], имеющая достаточно хорошую корреляцию с оценками людей. Метрика рассчитывается на основе пересечения n-грамм в автоматическом и эталонном переводах одного предложения. Для каждой n-граммы длины n рассчитывается доля P_n — среднее по всем предложениям в тестовом наборе отношения частоты n-граммы в кандидатах к частоте в эталонных переводах. При этом частота в кандидате ограничена значением в эталоне, чтобы отношение частот было ограничено сверху единицей:

$$P_n = \frac{\sum_{\text{n-gram} \in C} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{\text{n-gram}' \in C'} \text{Count}(\text{n-gram}')},$$

где C — множество предложений-переводов оцениваемой модели, C' — множество эталонных переводов тестового корпуса.

Далее BLEU рассчитывается как геометрическое среднее, умноженное на константу brevity penalty (BP)

$$BP = \min(e^{1-\frac{r}{c}}, 1),$$

$$BLEU = BP \sum_{n=1}^{4} \frac{1}{n} \log P_n,$$

где r — суммарная длина эталонных переводов тестового корпуса; c — суммарная длина переводов модели. Умножение на константу BP предлагается авторами метрики для поощрения более коротких переводов системы, так как более длинные тексты в среднем содержат больше случайных пересечений по n-граммам с эталонными текстами.

Метрика BLEU имеет хорошую корреляцию с человеческой оценкой качества переводов как при сравнении автоматических систем с профессиональными переводчиками, так и при сравнении автоматических систем друг с другом [7]. При проведении сравнения автоматических систем перевода, ав-

торы использовали модели различной сложности, а при сравнении профессиональных переводчиков выбирались эксперты с различным уровнем владения языками.

При проведении экспериментов метрика BLEU вычисляется по тестовым корпусам, подготовленным с помощью профессиональных переводчиков. Для экспериментов на англо-финском направлении использовались тестовые корпуса, подготовленные к конференции WMT-2017 [8]. Размер тестового корпуса для англо-финского составляет 1500 предложений, исходные предложения выбирались из новостных статей. Размер тестового корпуса для русско-казахского составляет 250 предложений, переведенных нанятыми профессиональными переводчиками.

3.3. Оценка качества циклического перевода

В данной работе в рамках улучшения качества перевода в целом предлагается улучшение согласованности переводов одного предложения прямой моделью и обратной ей. Для оценки улучшения согласованности необходимо выяснить, насколько сильно циклических перевод исходного текста искажает этот текст. С этим может помочь уже описанная метрика BLEU, позволяющая вычислять сходство между двумя текстами на одном языке. В данной работе метрику BLEU, рассчитанную для исходных текстов и их циклических переводов, будем называть CycleBLEU, и с ее помощью будем оценивать рост согласованности переводов прямой и обратной ей моделей.

В экспериментах ожидается, что после обучения с циклической функцией потерь (2) будет наблюдаться рост согласованности переводов, который можно будет увидеть с помощью метрики CycleBLEU, даже если прироста BLEU наблюдаться не будет. Это мотивируется тем, что рост согласованности прямой и обратной ей моделей сам по себе является достижением, даже при отсутствии заметного улучшения качества перевода на тестовых наборах.

4. Эксперименты

Теперь перейдем к подробному описанию экспериментов и условий их проведения. Кроме этого, опишем основные результаты, полученные при обучении модели перевода с циклической функцией потерь (2).

4.1. Описание процесса обучения

В проведенных экспериментах все модели перевода были построены на основе архитектуры Transformer. Исследование обучения с нуля с циклической функцией потерь (2) проводилось в конфигурации Transformer-tiny. В данной конфигурации размерность промежуточных векторных представлений составляет 256, а векторные представления внутри FFN блоков обладают размерностью 1024. Малый размер модели был выбран для изучения сходимости модели в условиях высокой дисперсии градиентов, так как в данных экспериментах финальное качество модели не является целью исследования.

В экспериментах с дообучением на циклическую функцию потерь (2) для инициализации моделей брались предобученные без циклической функции потерь (1) модели перевода конфигурации Transformer-base. В данной конфигурации размерность промежуточных векторных представлений составляет 512, а внутри FFN — 2048.

Во всех экспериментах оригинальная архитектура Transformer использовалась с механизмом относительного кодирования позиций [9]. Данный механизм кодирования позиций улучшает качество и уменьшает эффект переобучения под определенные позиции слов в предложении.

Как уже было описано, обучение проводилось с помощью метода Adam. Количество итераций нагрева было выбрано равным 2000 при дообучении. Во время итераций нагрева шаг градиента остается маленьким для накопления статистик, необходимых в алгоритме Adam. После нагрева размер шага обучения составляет 10^{-5}, после чего его значение уменьшается.

Все эксперименты с обучением проводились на вычислительном устройстве с 8 GPU Tesla M40. При этом при обработке пакета данных последний разделялся между всеми GPU, и вычисления проводились параллельно. В момент необходимости пересчета весов модели данные со всех вычислительных устройств объединялись и усреднялись для подсчета градиента.

4.2. Данные для экспериментов

Эксперименты проводятся на англо-финском и русско-казахском языковых направлениях. Обучающие пары текстов для данных экспериментов собраны с помощью обхода интернета и выделения интернет-страниц, имеющих переведенную версию сайта. После нахождения таких пар страницы преобразовываются в текстовый формат и выделяются параллельные предложения. Общее количество найденных таким образом пар предложений для каждого направления оказалось равным приблизительно 50 млн.

Каждый текст из обучения обрабатывается с помощью преобразования BPE [5]. Данное преобразование разделяет каждое слово на составные части некоторым выученным способом, уменьшая при этом общее количество уникальных слов в словаре модели. Данное преобразование помогает сократить количество параметров в модели и улучшает качество. Итоговый размер словаря после преобразования BPE для модели перевода составляет 32 000 токенов.

Кроме параллельных пар предложений в экспериментах с обучением используются синтетические пары. Такие пары можно получить путем перевода текста на целевом языке на входной язык с использованием обратной модели перевода. При использовании синтетических пар в обучении эффект от обучения с циклической функцией потерь (2) может уменьшиться, так как при обучении на таких данных модель уже и без циклической функции потерь получает информацию от обратной модели через синтетические переводы. Именно поэтому для экспериментов были взяты направления, на которых в

открытом доступе существует не так много качественных данных для целевого языка. Количество данных из News Crawl 2018 на целевом языке для создания синтетических пар на русско-казахском составляет 5 млн. предложений, что меньше, чем количество параллельных предложений в 10 раз. Для финского количество текстов на целевом языке составляет 50 млн. предложений, взятых из News Crawl 2014. Количество синтетических данных на англо-финском направлении сравнимо с числом параллельных пар предложений.

При обучении с синтетическими парами принято использовать данные на целевом языке большего объема, чем параллельные. Однако данные на целевом языке должны обладать высоким качеством, в противном случае синтетические примеры ухудшают качество переводных моделей. Для финского и казахского языков в открытом доступе не удалось найти большего количества качественных данных.

4.3. Эксперименты с обучением с нуля

При обучении с циклической функцией потерь (2) удалось получить формулы для вычисления градиентов, которые имеют общее с формулами градиентов из оригинальной статьи [2]. Однако важным отличием является то, что в предложенной функции потерь (2) не используются языковые модели. Хранение языковых моделей в памяти вычислительных устройств существенно ограничит количество свободных ресурсов, что приведет к уменьшению реального количества данных, обрабатываемых на одной итерации обучения. Уменьшение количества обрабатываемых данных на устройстве неизбежно приведет к заметному падению качества.

Кроме того, чтобы уменьшить расход памяти вычислительного устройства, будем оптимизировать только параметры прямой модели с помощью градиентов от циклических переводов (2), а в качестве обратной модели будем использовать уже предобученную. В описанных условиях обучение модели архитектуры Transformer-base с нуля составляет чуть менее двух недель.

Описанные ограничения вычислительных ресурсов делают применение оригинального метода [2] неэффективным для современных нейросетевых архитектур. Поэтому будем исследовать разработанный метод отдельно.

В первую очередь необходимо исследовать сходимость обучения с циклической функцией потерь (2) для современной модели перевода архитектуры Transformer. Для начала исследуем модель с небольшим числом параметров конфигурации Transformer-tiny. Обучение проводится на направлении с английского на финский на описанных параллельных данных с добавлением синтетических.

Качество обученных в течение 30 000 итераций моделей перевода представлено в табл. 1. Можно заметить, что качество обучения с использованием циклической функции потерь (2) значительно уступает обучению без циклических переводов (1). Все данные экспериментов приведены после подбора гиперпараметров обучения и представляют лучшие результаты для каждого

способа обучения. Кроме того, необходимо отметить и низкое значение CycleBLEU при обучении с циклической функцией потерь, хотя предложенный способ обучения должен растить данную метрику почти напрямую.

Таблица 1. Результаты при обучении с нуля на англо-финском направлении

Модель	BLEU	CycleBLEU
Прямая	12,0	50,0
Прямая+обратная	10,0	42,0

Данные результаты объясняются тем, что на первых этапах обучения градиенты, даваемые циклической функцией потерь (3), обладают крайне высокой дисперсией из-за того, что переводы, генерируемые прямой моделью, обладают низким качеством. Для таких переводов оценка обратной моделью является крайне шумной и не несет полезной информации для обучения. Решить данную проблему можно с использованием предобученной модели перевода для инициализации прямой модели в процедуре обучения с циклической функцией потерь (2), так как в таком случае переводы, генерируемые прямой моделью, с самого начала обучения были бы качественными.

4.4. Эксперименты с дообучением моделей

Для инициализации прямой модели в экспериментах с дообучением используется модель архитектуры Transformer-base, обученная до сходимости с функцией потерь без циклических переводов (1).

Данные для дообучения используются те же, что и во время предобучения и включают в себя как параллельные примеры, так и синтетические. Так как размер шага при обновлении весов в процедуре дообучения значительно меньше, качество модели на тестовых наборах во время дообучения может расти и без использования процедуры циклического перевода (2). По этой причине результаты обучения предложенного метода будем сравнивать как с предобученной моделью, так и с дообученной без использования циклических переводов (1).

Результаты дообучения для направления с английского на финский представлены в табл. 2. Качество дообучения с использованием циклической функции потерь (2) дает заметный прирост по BLEU. Таким образом, можно утверждать, что предложенный метод растит общее качество переводной модели даже по сравнению с обычным дообучением.

Таблица 2. Результаты дообучения с циклической функцией потерь для англо-финского направления

Модель	BLEU	CycleBLEU
Без дообучения	23,4	47,0
Прямая	25,0	55,0
Прямая+обратная	25,5	54,5

Результат данного эксперимента может показаться необычным, если сравнить результаты обучения с точки зрения метрики CycleBLEU. При обучении с циклической функцией потерь (2) качество получилось хуже, чем у обычного дообучения (1). Этот результат кажется неожиданным, так как предложенный метод должен напрямую увеличивать согласованность циклических переводов с изначальными текстами. Данный результат объясняется большой долей синтетических переводов в обучающей выборке. Как уже говорилось, использование синтетических данных в обучении, сгенерированных с помощью обратной модели, позволяет передать информацию в прямую модель из обратной еще до использования циклической функции потерь (2). Видимо, в данном эксперименте для высокого качества циклических переводов достаточно синтетических данных.

Рассмотрим теперь эксперименты с дообучением для направления с русского на казахский. Результаты эксперимента приведены в табл. 3. В данном эксперименте также наблюдается рост общего качества перевода, однако меньший, чем на англо-финском направлении.

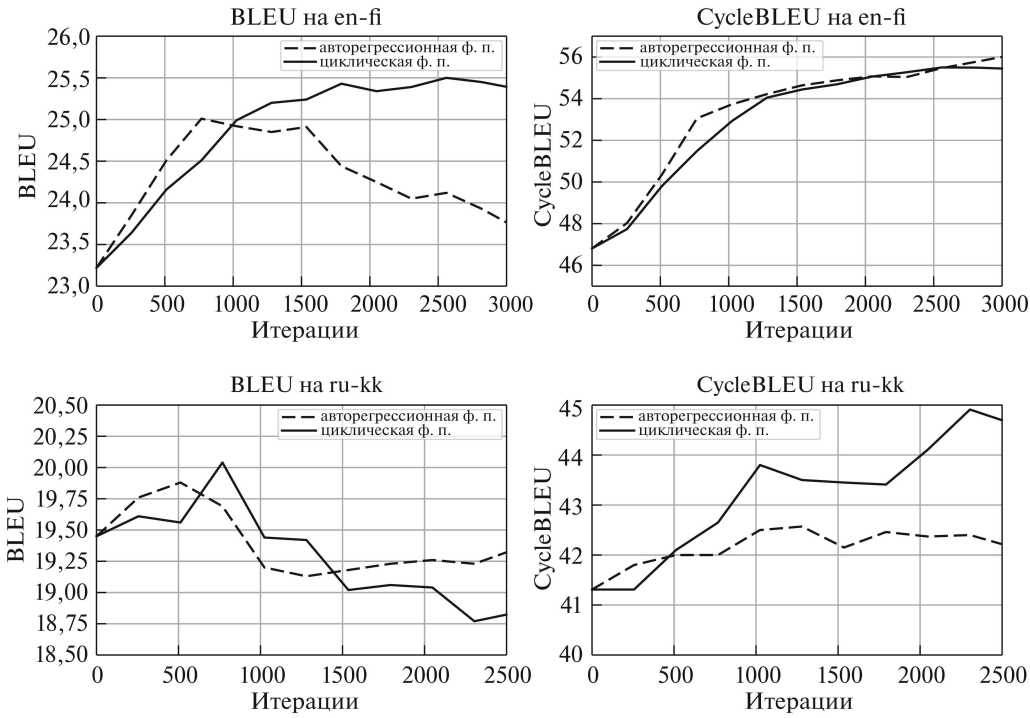
Таблица 3. Результаты дообучения с циклической функцией потерь для русско-казахского направления

Модель	BLEU	CycleBLEU
Без дообучения	19,5	41,5
Прямая	19,80	42,5
Прямая+обратная	20,05	44,9

Если обратить внимание на метрику CycleBLEU, то на данном направлении виден уже существенный прирост метрики при обучения с функцией потерь с циклическими переводами (2). Более заметный по сравнению с экспериментом на англо-финском направлении прирост CycleBLEU объясняется заметно меньшей долей синтетических примеров в обучающей выборке.

На рисунке представлены значения метрик BLEU и CycleBLEU на тестовых наборах в процессе обучения моделей на англо-финском и русско-казахском направлениях. В эксперименте с англо-финским направлением рост качества модели, обученной с циклической функцией потерь (2), больше, а график менее шумный. В то же время прироста по метрике CycleBLEU нет, что объясняется наличием большого количества синтетических примеров в обучающей выборке. На русско-казахском направлении наблюдается небольшой рост BLEU, но при этом виден заметный прирост CycleBLEU. Более высокий шум, который можно видеть на графиках русско-казахского направления можно объяснить малым размером тестового корпуса, который удалось собрать с привлечением профессиональных переводчиков.

В итоге общее число итераций, во время которых качество моделей улучшается при дообучении, не превосходит 3000, что в 10 раз меньше количества



Графики дообучения моделей с использованием циклической функции потерь и без. Верхняя строка соответствует направлению en-fi, нижняя — направление ru-kk. Первая колонка соответствует графикам BLEU, вторая — графикам CycleBLEU.

итераций, необходимых для обучения модели перевода с нуля. Таким образом, улучшения качества при использовании функции потерь с циклическими переводами (2) удается добиться при более быстром обучении, чем обучение с нуля, как в оригинальной работе [2].

В целом использование дообучения с циклическими переводами (2) позволяет улучшить качество переводов, хотя прирост оказался меньше, чем заявлялся в оригинальной работе [2]. При этом отдельно стоит отметить вычислительную эффективность предложенной процедуры дообучения, которая позволяет применить совместное обучение с обратной моделью в моделях перевода с современными архитектурами Transformer.

5. Заключение

В данной работе предложен подход для улучшения качества автоматических систем перевода с точки зрения согласованности с переводами обратной модели. Удалось показать, что использование процедуры обучения совместно с обратной моделью растит как общее качество перевода прямых моделей, так

и согласованность переводов обратной модели с исходными текстами. Последнее делает модель перевода более стабильной и уменьшает риск изменения смысла текста при использовании системы.

Вероятностная модель циклических переводов, используемая в ходе разработки метода обучения с обратной моделью, позволила получить формулы для оптимизации переводных моделей, схожие с полученными в работе [2], однако там не было представлено полных обоснований для используемых формул, а предложенные вспомогательные языковые модели делают процесс обучения современных моделей более трудоемким и менее стабильным. Предложенная модель дает пространство для развития и модификации подходов, использующих обратные модели перевода, путем изменения исходной вероятностной модели циклических переводов и добавления регуляризации.

СПИСОК ЛИТЕРАТУРЫ

1. *Stahlberg F.* Neural Machine Translation: A Review // J. Artific. Intelligence Res. 2020. No. 69. P. 343–418.
2. *Yingce Xia, Di He, Tao Qin, et. al.* Dual learning for machine translation // In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16). Curran Associates Inc., Red Hook, N.Y. 2016. P. 820–828.
3. *Bahdanau D., Cho K., Bengio Y.* Neural Machine Translation by Jointly Learning to Align and Translate // CoRR. 2015. abs/1409.0473.
4. *Kaplan J., McCandlish S., Henighan T., et. al.* Scaling Laws for Neural Language Models // arxiv.org
5. *Vaswani A., Shazeer N., Parmar N., et. al.* Attention is all you need // In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). 2017. Curran Associates Inc., Red Hook, N.Y., P. 6000–6010.
6. *Kingma D., Ba J.* Adam: A Method for Stochastic Optimization // 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, May 7–9, 2015. Conference Track Proceedings.
7. *Papineni K., Roukos S., Ward T., et. al.* Bleu: a Method for Automatic Evaluation of Machine Translation // Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. 2002. P. 311–318.
8. *Bojar O., Chatterjee R., Federmann C., et al.* Findings of the 2017 Conference on Machine Translation (WMT17) // Proceedings of the Second Conference on Machine Translation. 2017. Volume 2: Shared Task Papers.
9. *Shaw P., Uszkoreit J., Vaswani A.* Self-Attention with Relative Position Representations // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018. P. 464–468.

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 23.01.2022

После доработки 30.05.2022

Принята к публикации 29.06.2022