

Аннотация

Целью данной работы является разработка алгоритма переноса стиля из неформального стиля, присущего интернет общению, в стиль формальный и наоборот.

В ходе выполнения исследования был собран параллельный набор данных из 18 тысяч параллельных пар предложений неформальный-формальный стиль. В основе набора данных лежит корпус текстов статей с сайта Луркморье. Для разметки данных были привлечены студенты-лингвисты из Российского Государственного Гуманитарного Университета.

В работе были исследованы 4 различных метода, как основанные на полном отсутствии параллельных данных, так и с привлечением современных языковых моделей. Подходы, основанные на использовании больших языковых моделей, оказались наиболее эффективными для решения поставленной задачи. Данные методы, совместно с параметро-эффективными методами обучения, не требуют наличия большого количества параллельных данных, а с увеличением количества параметром языковой модели, объём необходимого количества данных для дообучения сводится до минимума.

Содержание

Введение	6
1 Анализ проблемы	7
1.1 Постановка задачи	7
1.2 Понятие стиля текста	7
1.3 Метрики качества	8
1.3.1 Выраженность стиля	8
1.3.2 Сохранение семантики текста	8
1.3.3 Плавность текста	10
1.3.4 Человеческая оценка	10
1.4 Наборы данных	11
1.5 Обзор существующих подходов	12
1.5.1 Редактирование ключевых слов	12
1.5.2 Состязательное обучение	14
1.5.3 Построение параллельного псевдо-корпуса	15
1.5.4 Генерация с помощью управления атрибутами	16
1.5.5 Манипуляции в латентном пространстве	17
1.5.6 Обучение с подкреплением	19
2 Создание набора данных	21
2.1 Обзор наборов данных, подходящих для аннотации	21
2.2 Разметка данных	22
2.2.1 Толока	22
2.2.2 Привлечение студентов из РГГУ	26
3 Проведение экспериментов	28
3.1 Используемые инструменты	28
3.2 Метрики качества	28
3.3 Обучение без учителя: Denoising Auto-Encoder	28
3.4 Guided Generation	32
3.5 Parameter-efficient fine-tuning	35
3.5.1 LoRA	35
3.5.2 P-Tuning	38
3.6 Few-Shot LLM	39
3.7 Результаты	40

4	Развитие работы	41
4.1	Продолжение создания набора данных	41
4.2	Улучшение метрик качества	41
4.3	Улучшение алгоритмов	41
	Заключение	42
	Список использованных источников	43
	Благодарности	48
A	Примеры генераций	49

Обозначения и сокращения

В настоящем отчете о НИР применяют следующие сокращения и обозначения

КОЛ-ВО — количество

РГГУ — Российский Государственный Гуманитарный Университет

VAE — variational auto-encoder

DAE — denoising auto-encoder

LLM — Large Language Models

GPT — Generative pre-trained transformer

PEFT — Parameter-Efficient Fine-Tuning

LoRA — Low-Rank Adaptation of Large Language Models

Введение

Возможность управлять стилем текста является очень актуальной и важной задачей, потому что это делает обработку естественного языка более ориентированной на пользователя. Алгоритмы переноса стиля текста имеют множество непосредственных применений. Например, одним из таких приложений являются интеллектуальные боты, для которых пользователи предпочитают эмоциональную и последовательную персону. Другим применением является разработка интеллектуальных помощников по написанию текстов; например, авторам-неспециалистам часто требуется улучшить свои тексты, чтобы они лучше соответствовали их назначению, например, более профессиональному, вежливому, объективному, с чувством юмора или другим продвинутым требованиям к письму, на освоение которых могут потребоваться годы опыта. Другие приложения включают автоматическое упрощение текста, увеличение объективности онлайн-текста, борьбу с оскорбительными выражениями и токсичными высказываниями и т.д.

Данная задача близка к другим задачам компьютерной лингвистики, например, машинному переводу или парафразу. Поэтому методы и экспертиза, накопленные в решении этих проблем, могут применяться и для неё.

Для выполнения поставленной цели, необходимо решить следующие задачи:

- проанализировать текущее состояние исследуемой области, существующие подходы к решению данной или схожей задачи, имеющиеся наборы данных;
- определить необходимые метрики и критерии оценки качества будущего алгоритма;
- собрать собственный набор данных;
- провести эксперименты для проверки работоспособности выбранных методов на собранном наборе данных;
- сделать выводы на основании полученных результатов;
- сформулировать планы для дальнейшего развития исследования.

1 Анализ проблемы

1.1 Постановка задачи

Задачей работы является разработка модели перевода текста формально-нейтрального стиля в стиль неформальный, присущий стилю интернет-общения, например, подобного в интернет-энциклопедии Луркморье. И наоборот: из неформального в формальный.

Свести эту задачу к классическим методам машинного перевода трудно, потому что не существует подобного параллельного корпуса на русском языке достаточного объёма и собрать его самостоятельно крайне сложно. Однако, параллельный корпус небольшого объёма (от 10 тысяч предложений) необходим. Во-первых, нужно оценить результаты работы будущей модели, например, с помощью BLUE score. Во-вторых, существуют методы обучения с учителем, не требующие больших объёмов данных и они будут исследованы в данной работе.

1.2 Понятие стиля текста

В рамках данной задачи можно выделить два определения стиля:

а) **Лингвистическое.** Стиль – это интуитивное понятие, которое включает в себя способ передачи некоторой семантики [1]. Семантика (или „содержание“) текста относится к предмету или аргументу, который автор хочет донести. Стиль текста – это литературный элемент, который описывает способы использования автором языка, включая выбор слов, структуру предложения, образный язык и расположение предложений, и работает вместе для создания тона, образа и семантики в тексте. Стиль указывает на то, как автор описывает события, объекты и идеи. С помощью стиля автор может предоставить дополнительную информацию, которую читатель сможет интерпретировать и на которую может реагировать. Невозможно перечислить все возможные стили. Каждый человек обладает уникальным набором приемов, часто адаптированных к конкретным людям, для достижения своих межличностных целей.

б) **Основанное на данных** (используемое в исследованиях по переносу стиля текста) – часто исследователи рассматривают „стиль“ как атри-

буты текста, зависящие от корпусов, принадлежащих конкретному стилю. Эти специфичные корпуса обычно содержат стилистические атрибуты, которые нетрудно смоделировать с помощью методов машинного обучения [2].

1.3 Метрики качества

Качество переноса стиля принято оценивать по трём параметрам: выраженность стиля (поменялся ли стиль в сгенерированном предложении, как этого требовалось), сохранение смысла (сохранилась ли семантика исходного предложения) и плавность текста (text fluency, насколько хорошо сгенерированное предложение соответствует грамматике и нормам языка) [3]. Помимо автоматических метрик для оценки указанных параметров, распространённым является использование человеческого труда для оценки качества.

1.3.1 Выраженность стиля

Для автоматической оценки выраженности стиля большинство исследователей обучают классификатор, который определяет имеет ли сгенерированное предложение желаемые стилистические атрибуты [2]. Общее качество определяется как соотношение количества правильно классифицированных предложений к общему числу предложений в тестовом наборе данных. Чем выше значение, тем лучше модель переносит стиль. Стоит отметить, что данная метрика не всегда показывает реальное качество алгоритма, так как оценки классификатора могут не коррелировать с человеческими оценками [4].

1.3.2 Сохранение семантики текста

Чтобы количественно измерить объем оригинальной семантики, сохраняемой после операции переноса стиля, применяются автоматические метрики, которые хорошо зарекомендовали себя в других задачах генерации естественного языка:

— BLEU [5], sacreBLEU [6] – изначально была создана для оценки качества машинного перевода. Это первая метрика, которая показала высокую корреляцию между человеческими оценками и качеством переведённого текста [7]. Для её подсчёта сгенерированный текст сравнивается с набором заранее подготовленных высококачественных референсов. Однако, у этой метрики есть определённый набор недостатков. Во-первых, в задаче переноса стиля текста простое копирование предложения может дать высокую оценку, потому что большое количество n -грамм часто переиспользуется в парафразе [2]. Во-вторых, некоторые исследования показывают, что метрика может показывать низкую корреляцию с человеческими оценками [4, 8]. В-третьих, обязателен параллельный набор данных для оценки, что не всегда имеется;

— source-BLEU/self-BLEU – в связи с тем, что параллельный набор данных часто недоступен, а метрика BLEU подразумевает его наличие, исследователи используют его модификацию, в которой переписанное предложение сравнивается с исходным предложением. Интуиция заключается в предположении, что семантика предложения сохраняется и перефразированное предложение будет иметь много совпадающих n -грамм с исходным предложением;

— METEOR [9] – оценивает сгенерированное предложение, сопоставляя его с одним или несколькими референсами. Сравнение основано на точном совпадении слов и фраз, основ слова (стемах), синонимах и перефразировках. Метрика рассчитывается как среднее гармоническое значение точности (precision) униграммы и полноты (recall), при этом полнота взвешивается выше;

— ROUGE-L [10] – метрика, ориентированная на полноту (recall), созданная для оценки суммаризации текста, в которой используется концепция самой длинной общей подпоследовательности (Longest Common Subsequence, LCS). Интуиция, лежащая в основе данной метрики, подразумевает, что чем длиннее LCS между двумя предложениями, тем больше они схожи. Оценка равна 1, когда два предложения равны, и 0, в противном случае;

— Cosine Similarity, BERTScore и пр. – вычисляется косинусная близость эмбеддингов между сгенерированным и референсным предложениями [11, 12]. Подразумевается, что эмбеддинги сгенерированного и референсного предложения должны иметь близкое расстояние, чтобы сохранить семантику;

— и другие.

1.3.3 Плавность текста

Плавность текста является основным требованием к генерациям на естественном языке. Для оценки плавности используется метрика перплексии, которая вычисляется с помощью языковой модели, предварительно обученной на желаемом наборе данных [13].

Однако, эффективность перплексии до сих пор остаётся спорным вопросом. Некоторые исследования [14] показывают её высокую корреляцию с человеческими оценками, в то время как другие [8] утверждают, что значительная корреляция отсутствует.

Перплексия, полученная с помощью языковой модели, может иметь следующие нежелательные свойства [2]:

— Склонность оценивать выше более короткие предложения, чем длинные;

— Имея одинаковое значение, менее частые слова будут оцениваться ниже, чем частые;

— Качество языковой модели очень сильно влияет на метрику. Оценки перплексии, зависят от учебного корпуса, архитектуры и конфигурации языковой модели, а также конфигурации оптимизации. Следовательно, генерации различных парафразеров должны оцениваться с помощью точно такой же языковой модели для объективного сравнения, что усложняет оценку качества.

1.3.4 Человеческая оценка

Существует несколько способов провести ручную оценку качества. Выделяют точечную и попарную оценки. В первом случае у ассессора запра-

шивается абсолютная оценка результатов генерации модели, а в последнем случае оценивается какая из нескольких генераций лучше или просится ранжирование. Так же человек помимо общей оценки, может отдельно оценить качество переноса стиля, качество сохранения семантики и плавность текста.

Однако, человеческая оценка дорога и трудновоспроизводима. Использование ручного труда требует большого количества времени и финансовых вложений. Помимо этого, человеческие оценки, полученные в разных исследованиях часто нельзя сравнить напрямую, потому что они могут быть субъективными и невоспроизводимыми [15]. Некоторые стили очень сложно оценить без экспертизы и предварительной подготовки.

1.4 Наборы данных

Задача переноса стиля достаточно обширна. Выделяют следующие подзадачи: изменение формальности, вежливости, токсичности, сентимента и пр. Данная работа фокусируется на изменении формальности текста, поэтому рассмотрению подлежат существующие наборы данных для этой задачи.

Изменение формальности это один из наиболее отличительных стилистических аспектов, который можно наблюдать во многих лингвистических явлениях, таких как полные имена вместо аббревиатур или сокращений, присутствие сленга и просторечий и пр. На текущий момент существует лишь несколько параллельных наборов данных для этой задачи. Самые популярные из них это англоязычный Grammarly's Yahoo Answers Formality Corpus (GYAFC) [16] и мультиязычный (Итальянский, Французский и Португальский) XFORMAL [17]. К сожалению, на русском языке отсутствуют параллельные наборы данных, состоящие из пар формальное-неформальное предложение. Поэтому набор данных придётся собирать самостоятельно.

В таком случае, это создаёт другую задачу: как правильно организовать процесс разметки данных? Это достаточно нетривиальная задача. Сам процесс определения стиля уже написанного текста не имеет строгого детерминированного решения. Само понятие неформального стиля не явля-

ется строго определённым и тексты разной стилистической окраски могут подпасть под понятие неформального текста. А перефразирование человеком текста в неформальный стиль не является однозначным и зависит от навыков и кругозора человека, выполняющего задачу. Другая ситуация с формальным стилем – это литературный русский язык, встречающийся в новостях, документах и пр. В рамках данной работы делается предположение, что любой асессор с должным уровнем образования, владеет этой стилистикой. Переписать неформальный текст в формальном стиле является более простой и объяснимой задачей, нежели обратная. Поэтому таким образом и необходимо ставить задачу асессорам, которые будут заниматься созданием набора данных. В связи с вышесказанным, для разметки необходимо использовать русскоязычный корпус, выдержанный в неформальной стилистике и имеющий достаточный объём.

1.5 Обзор существующих подходов

Существующие алгоритмы могут быть классифицированы в зависимости от того, содержит ли набор данных параллельный текст с разными стилями или несколько непараллельных корпусов в одном стиле.

Большинство методов, подразумевающих наличие большого параллельного набора данных, используют стандартную нейросетевую модель sequence-to-sequence (seq2seq) с архитектурой энкодер-декодер, которая изначально была создана для решения задач машинного перевода [2].

Однако, так как получение параллельного корпуса большого объёма не представляется возможным, то в данной работе интерес представляют методы, основанные на обучении без учителя или с минимальным использованием параллельного набора данных.

1.5.1 Редактирование ключевых слов

Стилистические атрибуты текста, такие как формальность и сентимент, часто выражаются отдельными ключевыми словами и фразами. Например, слова „молодой человек“ и „телевизор“ имеют высокую степень формальности, а слова „чувак“ и „телик“ крайне неформальны и простореч-

ны. Соответственно, интуитивным решением будет замена специфических слов и выражений на их противоположные пары.

Показательным примером подхода по замене ключевых слов является работа *Delte-Retrieve-Generate* [4]. Имея текст в исходном стиле (например, текст с негативным сентиментом), модель сначала определяет стилистически ключевые слова, такие как „красивый“, „плохой“, путем подсчета частоты каждого слова. Основная интуиция заключается в том, что слова, приписываемые стилю, скорее всего, будут часто использоваться в предложениях определенного стиля. Затем модель удаляет из текста слова, приписываемые стилю. Предполагается, что итоговый текст на данном этапе содержит только содержательную информацию. Далее модель извлекает текст, аналогичный исходному тексту по смысловой нагрузке. Текст будет извлечен из целевого корпуса стилей (т.е. текстового корпуса с положительным сентиментом, в данном примере). Затем модель извлекает стилистические слова, из найденного текста, используя аналогичный подход. Наконец, слова, относящиеся к стилю извлеченного текста, объединяются со словами исходного текста для создания текста в целевом стиле с использованием подхода, основанного на правилах или модели Seq2Seq.

Преимущество подхода с заменой ключевых слов заключается в его простоте: модели относительно менее сложны и требуют меньшего времени обучения. Кроме того, перестановка ключевых слов и фраз также обеспечивает некоторую объяснимость модели: имеется возможность явно наблюдать, какая часть текста изменена для изменения стиля.

Однако, есть у подхода и недостатки. Во-первых, явная замена ключевых слов по большей части ограничена изменением сентимента текста и не может быть применена к другим задачам, таким как перенос формальности. Это связано со спецификой задачи переноса сентимента, где зачастую сентимент определён определённым лексиконом и его замена это рабочий механизм трансфера. С другой стороны перенос формальности выходит за пределы простой замены слов, потому что формальность может выражаться синтаксисом. Во-вторых, сложно применить замену ключевых слов к неполярным стилям, не таким как положительный-негативный и пр.

1.5.2 Состязательное обучение

Состязательное обучение (Adversarial Learning) – популярный подход, применяемый во многих методах для неявного разделения стиля и семантики [7]. В целом, состязательное обучение используется для двух целей:

- а) генерация текста, неотличимого от реального;
- б) удаление стилистических атрибутов в латентном представлении текста.

Репрезентативной работой в данном направлении является *Style Transfer in Text: Exploration and Evaluation* [11]. В ней предлагаются два подхода: с использованием нескольких декодировщиков и с использованием стилистических эмбеддингов (см. рисунок 1 слева и справа соответственно). В обеих моделях кодировщик обучается создавать промежуточное латентное представление для входной текстовой последовательности. Состязательная сеть используется чтобы разделить в этом представлении семантику от стиля.

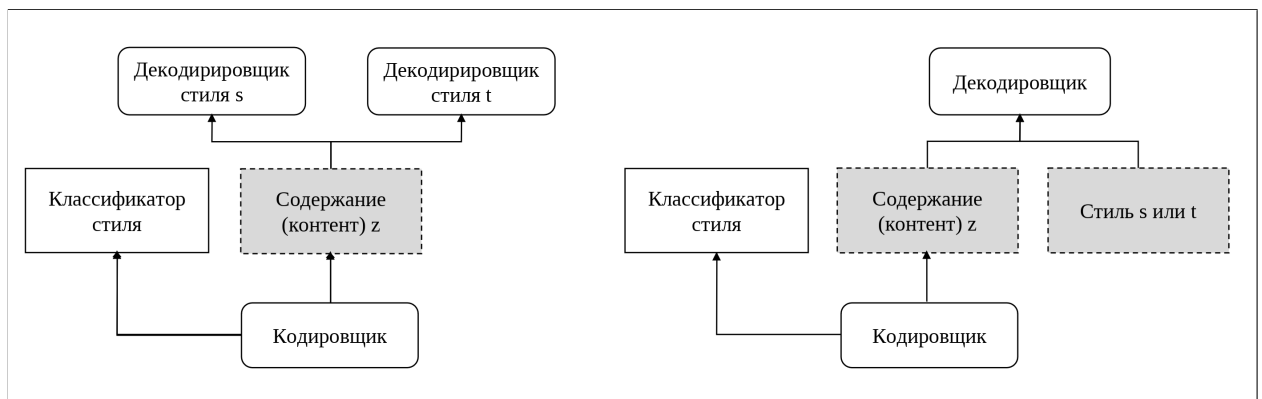


Рисунок 1 — Состязательное обучение. Слева: подход с использованием нескольких декодировщиков. Справа: подход и использованием стилистических эмбеддингов

Сама состязательная сеть состоит из двух главных компонент. Первый компонент (дискриминатор) предназначен для классификации стиля входной последовательности x на основе его латентного представления, созданного кодировщиком. Функция потерь, которую необходимо минимизировать, – это отрицательная логарифмическая вероятность меток стилей в

обучающих данных:

$$L_{adv1}(\Theta_D) = - \sum_{i=1}^M \log p(l_i | \text{Encoder}(x_i; \Theta_E); \Theta_D)$$

где Θ_D и Θ_E это параметры дискриминатора и кодировщика соответственно. M обозначает размер обучающего набора данных, а l_i обозначает метку стиля.

Второй компонент предназначен для обмана дискриминатора, чтобы тот не мог правильно идентифицировать стиль входной последовательности x . В этом случае функция потерь представляет собой максимизацию энтропии предсказанных меток стиля:

$$L_{adv2}(\Theta_E) = - \sum_{i=1}^M \sum_{j=1}^N H(p(j | \text{Encoder}(x_i; \Theta_E); \Theta_D))$$

где N - количество стилей. Обе части состязательной сети обновляют разный набор параметров. Они работают вместе, чтобы гарантировать, что выход $\text{Encoder}(x_i; \Theta_E)$ не содержит стилистическую информацию.

Как только кодировщик обучен создавать латентное представление, два генеративных подхода генерируют текст в целевом стиле. Первый подход подразумевает обучение нескольких декодеров под каждый стиль. Вторым подход использует обучение стилистических эмбедингов и конкатенации их с представлением содержимого для генерации текста в целевом стиле с помощью декодера.

1.5.3 Построение параллельного псевдо-корпуса

Обратный перевод (Back-Translation) изначально использовался в задаче машинного перевода для создания искусственного обучающего корпуса [18]. Затем данный подход был успешно использован в задачах переноса стиля. Выделяют два основных способа, основанных на поиске и генерации данных.

Первым из способов построения псевдопараллельных данных является поиск, а именно извлечение выровненных пар предложений из двух

непараллельных корпусов в разных стилях. В *Unsupervised text attribute transfer via iterative matching and translation* [19] эмпирически показано, что семантически сходные предложения в двух стилистически разных корпусах, как правило, являются аналогами друг друга, отличающимися атрибутами стиля. Следовательно, в работе строят изначальный псевдо-корпус путем сопоставления пар предложений в соответствии с косинусным сходством предварительно обученных эмбедингов предложений. Для каждого предложения x , его псевдо-аналог \hat{x}' это самое похожее предложение в другом корпусе X' :

$$\hat{x}' = \arg \max_{x' \in X'} \text{Similarity}(x, x')$$

Второй способ является генеративным, такой как итеративный обратный перевод (Iterative Back-Translation, IBT) в работе *Iterative back-translation for neural machine translation* [20]. Прежде чем начать итеративный процесс IBT должен инициализировать две модели переноса стиля: $M_{a \rightarrow a'}$, переводящая из стиля a в стиль a' , и $M_{a' \rightarrow a}$, делающая обратное. Затем на каждой итерации выполняется следующая последовательность шагов:

- а) Используются имеющиеся модели для генерации псевдо-параллельного корпуса. $M_{a \rightarrow a'}(x)$ генерирует псевдо-пары (x, \hat{x}') для $\forall x \in X$, а $M_{a' \rightarrow a}(x)$ генерирует (\hat{x}, x') для $\forall x' \in X'$;
- б) Повторно обучаются эти две модели переноса стиля на наборах данных, сгенерированных на первом шаге. То есть обучаются $M_{a \rightarrow a'}(x)$ на парах (\hat{x}, x') , а $M_{a' \rightarrow a}(x)$ на (x, \hat{x}') .

1.5.4 Генерация с помощью управления атрибутами

Этот метод подразумевает использование кода атрибута a для управления генерацией текста в различных стилях.

Функция потерь, управляемая классификатором, используется для гарантии того, что генератор G сгенерирует предложение x' с требуемым атрибутом стиля. А именно, функция потерь минимизирует:

$$L_{cls}(\Theta_G, t) = -\mathbb{E}_{p(x)}[\log D(x')]$$

где D это классификатор стиля (дискриминатор), обученный на данных x .

Так как метод генерации с помощью управления атрибутами сильно зависит от классификатора стиля, этот подход имеет такие же недостатки, как и методы состязательного обучения. В частности, точность классификатора стилей ограничивает выучивание качественных стилистических атрибутов.

В работе *Toward controlled generation of text* [21] используется вариационный автокодировщик (variational auto-encoder, VAE) [22], который выучивает латентное пространство z , и классификатор стиля, чтобы выучить вектор стилистических атрибутов a . Для этого используется следующая функция потерь:

$$L_{VAE}(\Theta_G, \Theta_E; x) = KL(q_E(z|x)||p(z)) - E_{q_E(z|x)q_D(a|x)}[\log p_G(x|z,a)]$$

где $KL(\cdot||\cdot)$ это дивергенция Кульбака-Лейблера, Θ_E и Θ_G – параметры энкодера и декодера соответственно. Условный вероятностный кодировщик E , обозначаемый как $q_E(z|x)$, выводит латентное представление z по заданному входному предложению x . $q_D(a|x)$ – это условное распределение, определенное классификатором D для каждой структурированной переменной в a . Общая схема модели проиллюстрирована на рисунке 2.

Чтобы гарантировать, что z сохраняет только информацию о семантике, не зависящую от стиля, предлагается ограничение независимости, гарантирующее, что латентное представление z входного предложения x и переданное предложение стиля x' остаются близкими друг к другу. Добавленное ограничение независимости, по существу, гарантирует, что информация о содержимом отделена от входного текста и закодирована в латентном представлении z .

1.5.5 Манипуляции в латентном пространстве

Другим подходом к решению задачи переноса стиля текста является внесение изменений в латентное пространство, полученное от обучения модели автокодировщика. Рисунок 3 иллюстрирует этот метод.

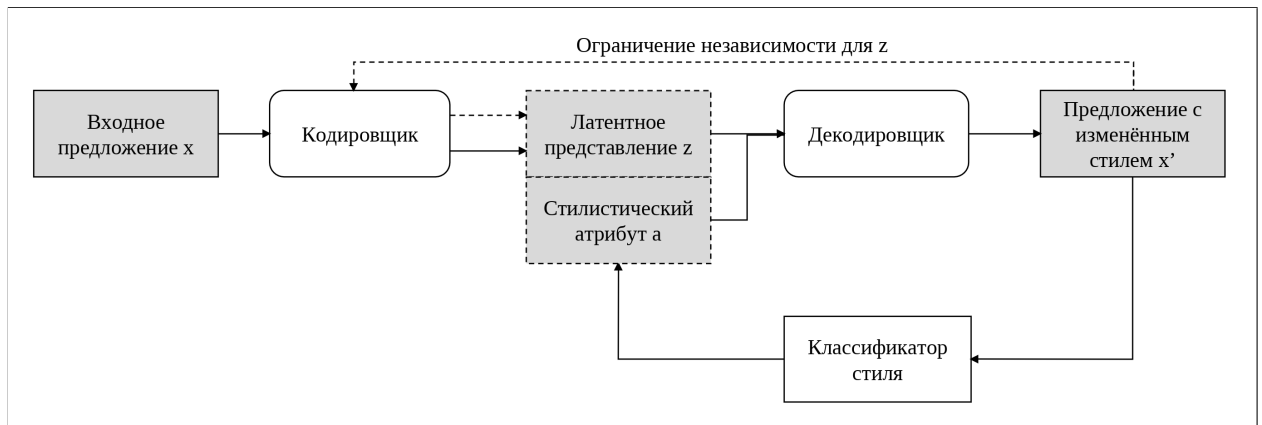


Рисунок 2 — Использование VAE для редактирования латентного пространства для трансфера стиля текста

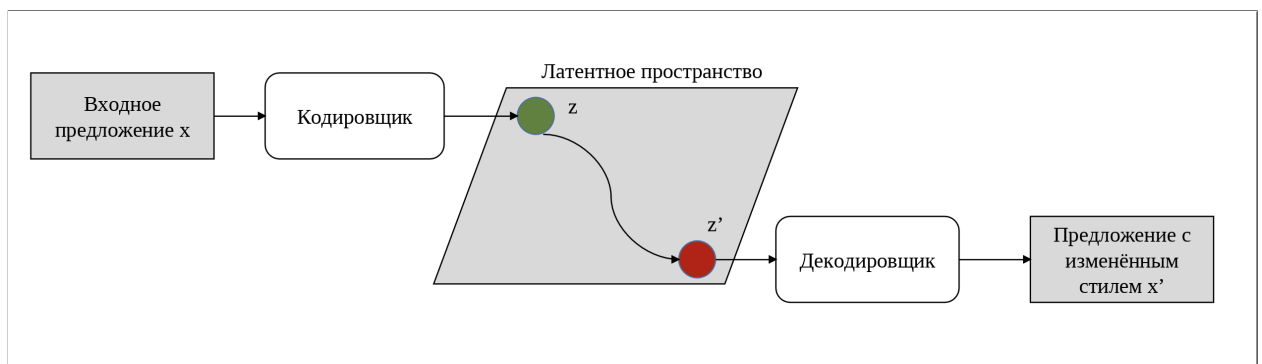


Рисунок 3 — Общая структура редактирования латентного пространства для трансфера стиля текста

Классификатор стиля обучается совместно с автоэнкодером. Во время обучения итеративно обновляется латентное пространство z в ограниченном пространстве, чтобы максимизировать точность классификатора. А именно, каждое обновление вычисляется на основе градиента потери классификатора стилей относительно z . Обработанное латентное представление z' затем подаётся в декодер для генерации текста целевого стиля.

Главной проблемой методов этого семейства является определение границ, в пределах которых должна происходить манипуляция в латентном пространстве. Было замечено, что декодер не может генерировать предложения в целевом стиле, если полученное после редактирования латентного пространства выходит за пределы областей представления, увиденных декодером во время обучения [23]. Существующие решения направлены на ограничение манипуляций в рамках ограниченного латентного пространства.

1.5.6 Обучение с подкреплением

Ключевая идея алгоритмов, основанных на обучении с подкреплением, это использование специально разработанных функций вознаграждения для управления процессом переноса стиля вместо различных функций потерь, используемых в других методах.

Алгоритм *Policy Gradient* [24] используется для максимизации ожидаемого вознаграждения за перефразированный текст. Однако из-за высокой дисперсии градиента выборки обучение с помощью этого метода может быть нестабильным.

В работе *A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer* [25] предложено обучать две Seq2Seq модели между двумя стилями посредством обучения с подкреплением, не разделяя стиль и семантику. Авторы рассматривали обучение модели переноса стиля из исходного в целевой и из целевого в обратный как двойную задачу. Награда за классификатор стилей R_S и награда за реконструкцию R_C предназначены для повышения точности передачи стиля и сохранения семантики. Общая награда является гармоническим средним из двух вознаграждений, и она использовалась в качестве сигнала обратной связи для обучения в структуре с двумя задачами. Таким образом, модель может быть обучена с помощью обучения с подкреплением без параллельных данных. Общая схема обучения представлена на рисунке 4

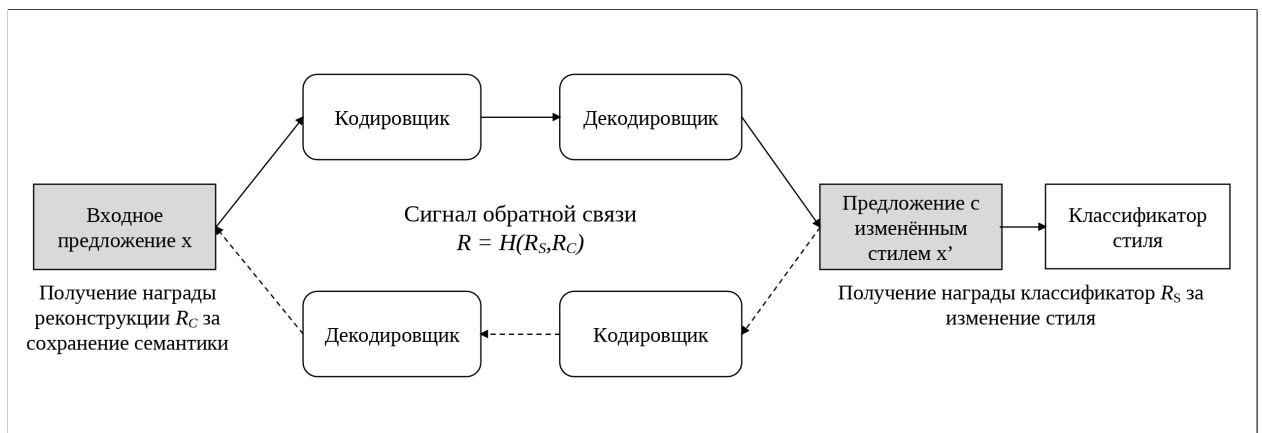


Рисунок 4 — Модель двойного обучения с подкреплением

Хотя существующие методы обучения с подкреплением по-прежнему в значительной степени опираются на классификаторы стилей для управле-

ния процессом переноса стиля, этот подход дает возможность разработать другие функции вознаграждения, которые будут направлять процесс переноса стиля.

2 Создание набора данных

На русском языке не существует набора данных, подходящего для целей исследования. Поэтому придётся создавать его самостоятельно. Для этого потребуется определить два корпуса: первый, состоящий из текстов в формальном стиле, и второй, состоящий из текстов в неформальном стиле. Затем, согласно главам 1.1 и 1.4, для части предложений из неформального корпуса необходимо создать их формальные парафразы.

2.1 Обзор наборов данных, подходящих для аннотации

Можно определить доступные корпуса на русском языке: Википедия, новостные издания, литература, печатные издания, субтитры, социальные сети, специализированные наборы данных для конкретных задач (например, детоксификация текста).

Социальные сети на первый взгляд подходят для создания неформального корпуса, однако стилистические атрибуты в них крайне неоднородны и значительно различаются у разных пользователей. Так же большое количество пользователей использует достаточно нейтральный стиль в своих комментариях.

Интересным кандидатом для использования в качестве источника неформального текста является база статей сайта Луркморье¹ (Lurkmore) за 2021 год. Луркморье – русскоязычная интернет-энциклопедия, аналог Википедии, функционирующая на том же движке MediaWiki, но написанная в неформальном стиле, изобилующая жаргонами и интернет-мемами, и позиционирующая себя как «энциклопедия современной культуры». Данный сайт, помимо того, что удовлетворяет критерию неформального стиля, является стилистическим антиподом Википедии. Также база статей имеет достаточный объём (около 9 тысяч статей, состоящих из более 600 тысяч предложений различной длины) для использования в обучении современных нейросетевых алгоритмов.

В данной работе выбор был сделан в пользу пары Луркморье-Википедия. Главной причиной такого решения является жанровая близость двух

¹Оригинальный сайт удалён владельцем. Обновлённая версия: <https://neolurk.org/wiki/Луркморье>

этих корпусов (здесь под жанром подразумевается функциональная направленность текста, как в работе *Functional Text Dimensions for annotation of Web corpora* [26]). Тексты из обоих корпусов являются энциклопедическими статьями, информирующими читателя о объектах действительности и явлениях.

Для разбивки текста на предложения был использован токенизатор Razdel от проекта Natasha¹. Затем принимались только предложения, соответствующие следующим параметрам:

- длина от 80 символов
- минимум 5 слов в предложении
- обязательно должна присутствовать кириллица.

Из базы можно получить около 423 тысяч предложений, что соответствует 125 МБ текста. Таким образом в итоговом корпусе оказалось 423974 предложения.

2.2 Разметка данных

2.2.1 Толока

Как было упомянуто в 1.1 10000 пар предложений формальный-неформальный стиль являются желаемым результатом. Такое в одиночку не сделать, поэтому необходимо воспользоваться помощью ассессоров. И подходящим сервисом для решения этой задачи является сервис Толока² – платформа для привлечения ассессоров к разметке данных.

Основная сложностью в процессе привлечения сторонних людей является описание задания, написание инструкции и фильтр подходящих кандидатов [27].

Для проверки работоспособности подхода было принято решение запустить тестовое задание на 100 случайных предложений из выборки. На рисунках 5 и 6 представлена инструкция, выдаваемая ассессорам. В ней пользователю Толоки объясняется, что нужно перефразировать предложения из сленга в формальный стиль. Дается пример источников текста

¹<https://github.com/natasha/razdel>

²<https://toloka.ai/>

в формальном стиле. Затем приводится несколько примеров перефразов. В инструкции также указывается предупреждение, что в задании может содержаться оскорбительный контент. Также толкеру сообщается, что неформальность заключается не только в наличии сленга, но и в стилистике предложения. Приведены несколько примеров. Указывается, что в случае невозможности перевести предложение или в случае, если толкер считает его уже формальным, необходимо его пропустить

В данном задании Вам необходимо перефразировать предложения, содержащие сленговые выражения, ненормативную лексику, а также различные интернет-мемы в формальный стиль

Что такое формальный стиль?

В формальном стиле написаны тексты документов, газет, википедии и т.п.

Например:

- "Завоевала огромную популярность у школоты и быдла по банальной причине наличия over 9000 готовых кривых шаблонов, халявных ущербных расширений и простой админки, не требующей наличия мозга."

Можно перевести как:

"Завоевала огромную популярность у молодой и непрофессиональной аудитории по причине наличия большого количества готовых шаблонов, бесплатных расширений и простой панели администратора, не требующей специализированных знаний"

- "Не вызывает сомнения, что жителей Нерезиновска это не интересует, а замкадышам хоть какое-то благо цивилизации."

"Не вызывает сомнения, что жителей Москвы это не интересует, а жителям области хоть какое-то благо цивилизации"

- "Зато процессор оверклокнут настолько, что телефон превращается в портативную жаровню."

"Зато частота процессора увеличена настолько, что телефон начинает очень сильно греться"

- "Алсо, бравый кадет засветился на немногочисленных, но порой доставляющих фотожабах, во многом благодаря своему лику."

"Также бравый кадет был замечен на немногочисленных, но очень смешных карикатурах, во многом благодаря своему внешнему виду."

Внимание! Текст может содержать абсценную лексику, а так же оскорбления различного характера! Если какое-то предложение вы посчитаете оскорбительным, то его можно таковым пометить и мы постараемся сделать всё, чтобы избежать подобных ситуаций в дальнейшем.

Рисунок 5 — Первая часть инструкции для разметчиков

Само задание представляет лишь обязательное поле, в которое ассессор должен записать парафраз. Изображение интерфейса задания представлено на рисунке 7. На одну страницу давалось одно задание, чтобы ассессор мог на нём сфокусироваться. Временной лимит устанавливался в 10 минут.

Так же было установлено тридцатипроцентное качество исполнителей. Установлено перекрытие равное 2, т.е. каждое предложение продуб-

Формальность заключается не только в наличии сленга, но и в стилистике предложения

- И мрут эти солдаты, как мухи, ибо Эрвин готов пожертвовать многим ради спасения всего остального человечества.

Среди этих солдат очень большие потери, потому что Эрвин готов пожертвовать многим ради спасения всего остального человечества.

- Был запущен в 2014 изначально для Индии, ибо гендиректор Google Сундар Пичаи ВНЕЗАПНО индус.

Был запущен в 2014 году изначально для Индии, возможно потому, что гендиректор Google Сундар Пичаи - индус.

- 42-я серия небезызвестного "Том и Джерри" ВНЕЗАПНО повествует о смерти, рае, аде, жестокости и прощении - наполнена СПГС и психоделическими картинками чуть менее, чем полностью.

42-я серия небезызвестного "Том и Джерри" неожиданно повествует о смерти, рае, аде, жестокости и прощении - наполнена так называемым "синдромом поиска глубинного смысла" и большим количеством психоделических картин.

- Закончилось это всё массовым баном даже тех IP-шников, с которых на Апачан ни разу не выходили.

Закончилось это всё массовой блокировкой даже тех IP-адресов, с которых на Апачан ни разу не выходили.

Если предложение перевести не получается, или Вам кажется что оно уже написано в формальном стиле - пропустите его.

Рисунок 6 — Вторая часть инструкции для разметчиков

Рисунок 7 — Окно задания для разметчика

лируется двум разным исполнителям. И, самое главное, была установлена отложенная приёмка, что означает, что ассессоры получают деньги, лишь при явном одобрении результата. Полный список настроек представлен на рисунке 8.

Получились следующие результаты. Для 100 предложений с перекрытием 2 было создано 200 заданий. После ручной проверки были приняты 96 перефразов, 44 был отклонены и с 60 ответами возникли трудности. Постановку задачи можно считать удовлетворяющей.

Однако, у многих ассессоров возникли некоторые трудности:

— Грамматические ошибки. Исполнитель либо опечатался, либо использовал неправильно причастие или спряжение, таких случаев немного. Такие предложения можно принять, потому что эти ошибки можно легко выявить и исправить.

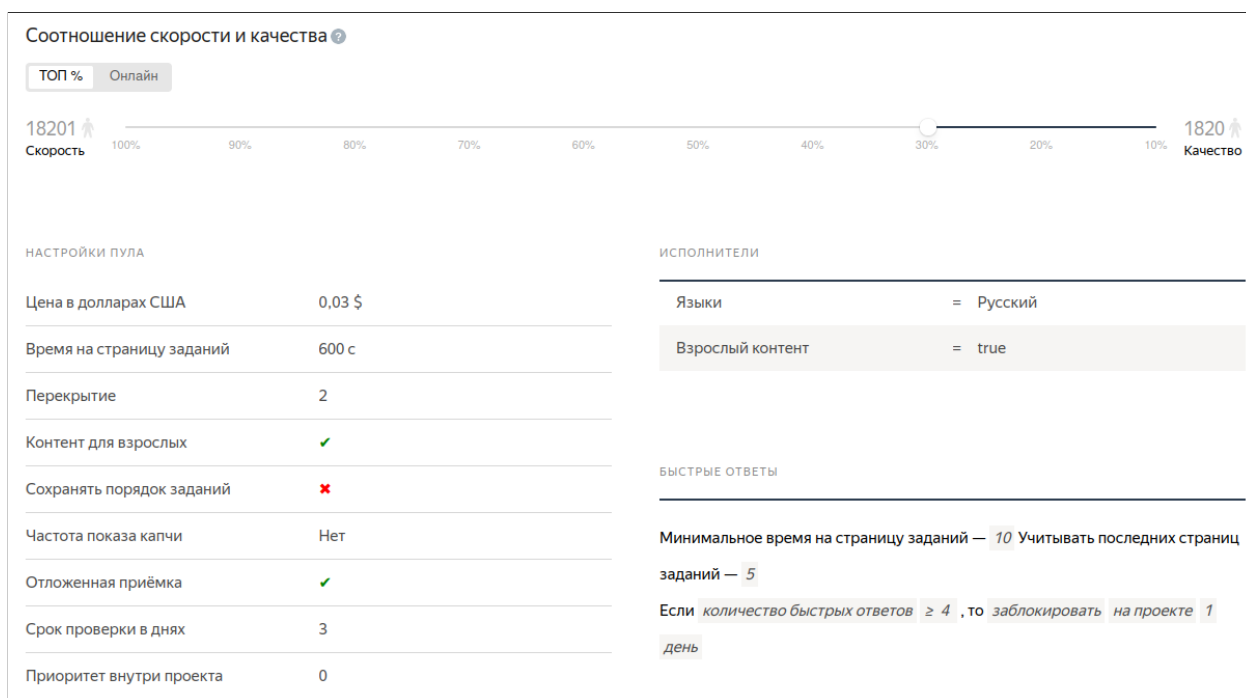


Рисунок 8 — Параметры задания

— Формализована лишь часть предложения. Распространённый случай, когда, например, исполнитель заменил обценное слово на нейтральное, но в тоже время не затронул другие стилистически неформальные конструкции.

— Исполнитель не понял сленга. Здесь исполнитель либо не распознал отсылку на интернет-мем, либо неправильно её понял.

— Искажён смысл. Ошибка зачастую связанная с предыдущей, когда асесор очевидным образом изменил посыл предложения.

— Также распространены случаи, когда исполнитель вроде перефразировал, выполнил задачу, но стилистически текст всё ещё несёт в себе просторечия.

Сразу появилось много идей, как можно улучшить качество:

- Добавить обучение
- Увеличить соотношение качество/скорость
- Увеличить перекрытие (затратно)
- Запускать разметку по частям и отфильтровывать исполнителей
- Расписать более подробно критерии приёмки
- Уменьшить длину предложений

Также необходимо оценить финансовые затраты. В среднем ассессор тратил около двух минут на выполнение задания (немного медленнее, чем в [27]), соответственно он может выполнять 30 заданий в час. Если установить цену в \$0,02 за задание, то один ассессор обходится в \$0,6/час. Тогда необходимые 10000 предложений обойдутся в \$200, что по курсу 80 рублей за доллар будет составлять 16000 рублей. Данная цена является достаточно большой.

2.2.2 Привлечение студентов из РГГУ

В связи с этими проблемами было рекомендовано обратиться к студентам-лингвистам из Российского Государственного Гуманитарного университета и попросить помочь их с разметкой данных. Спасибо Александре Ивойловой и её студентам, что откликнулись и начали помогать с выполнением работы.

В создании набора данных приняли участие 16 студентов старших курсов. Им были объяснена специфика и проблемы, возникшие на Толоке.

Целью было поставлено собрать 10000 перефразированных предложений. Для оценки качества работы алгоритма этого будет достаточно.

Для выполнения задачи Александра Ивойлова и я создали небольшую программу, которая автоматически сохраняла результаты и различные метрики. Интерфейс программы представлен на рисунке 9.

Вверху указано переводимое предложение, а в нижнем окне студент должен записать свой парафраз. Так же, если предложение вызывает трудности у студента, то его можно соответствующим образом пометить. Это сделано для того, чтобы можно было в дальнейшем фильтровать предложения. Если оригинал содержит артефакты разметки, то можно его исправить после разметки и получить хорошее предложение. Артефакты разметки означают, что само исходное предложение качественное, но в нем остались артефакты от вики-движка и нужно вручную их почистить.

Аннотирование длилось четыре месяца и были получены следующие результаты:

— Всего обработано 30118 предложений;

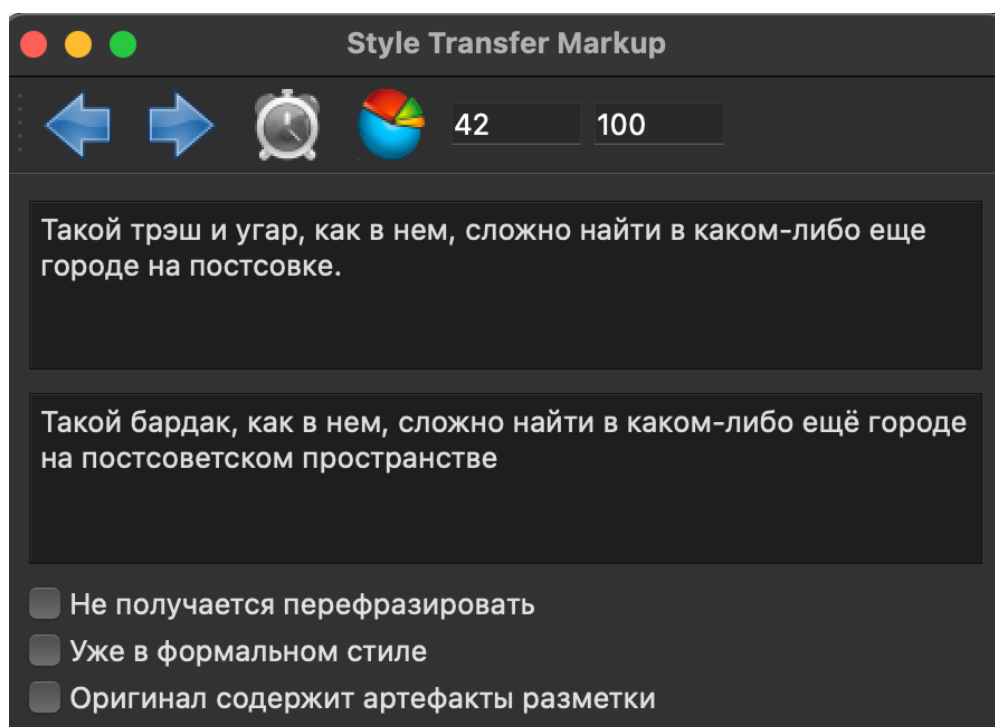


Рисунок 9 — Программа разметки данных для студентов РГГУ

— 17881 предложение готовы к использованию;
— 1271 одно предложение были помечены как имеющие артефакты.
После исправления артефактов и фильтрации осталось 659 предложений.

Итого получилось 18 540 предложений. Поставленная цель перевыполнена.

Студенты перевели большую часть неформальностей, интернет-мемов и отсылок, а те, которые они не понимали, искали им объяснения в интернете и пытались сделать хороший перевод. В целом эту работу можно оценить как положительную.

Однако, некоторые аспекты можно было выполнить лучше. Со стороны студентов: несмотря на просьбы пропускать очевидно формальные предложения, часто студенты их не игнорировали и переписывали с минимальными изменениями. С нашей стороны необходимо было более качественно подходить к фильтрованию набора данных и инвестировать больше времени в поиск более качественного алгоритма по формированию выборки.

3 Проведение экспериментов

3.1 Используемые инструменты

Разработка велась на языке программирования *Python 3.8*. Реализация нейронных сетей и оценка качества выполнены с помощью библиотек *PyTorch* [28], *transformers* и *evaluate* от проекта *HuggingFace* [29]. Предобученные модели получены из *HuggingFaceHub*, открытого репозитория моделей и наборов данных. Обучение выполнялось на видеокарте Nvidia T4.

3.2 Метрики качества

Для оценки качества алгоритма будут использоваться следующие метрики:

- SacreBLEU [6];
- METEOR [9];
- Style Score – дообученный с помощью параметро-эффективного метода LoRA [30] на собранном наборе данных классификатор стиля. В качестве изначальных весов взят *xlm-roberta-base*. Качество классификатора: Accuracy: 86%, F-1: 86%, Precision: 91%, Recall: 83%;
- Semantic Score – дообученный с помощью параметро-эффективного метода LoRA [30] на собранном наборе данных классификатор семантики, определяющий как сильно сохраняют два предложения одно и то же содержание. В качестве изначальных весов взят *cointegrated/rubert-tiny2*. Качество классификатора: Accuracy: 85%, F-1: 86%, Precision: 91%, Recall: 82%.

3.3 Обучение без учителя: Denoising Auto-Encoder

Генерация с помощью управления атрибутами с использованием архитектуры denoising auto-encoder [31, 32, 33] является распространённым решением в ситуации отсутствия параллельных данных.

В данном подходе были выработаны три принципа для решения задачи. Визуализация этих принципов представлена на рисунке 10. На изображении под пунктом А показаны исходные данные: два набора данных,

каждый со своим стилем (разные цвета и точки соответствуют предложениям с соответствующим стилем). Под пунктом Б представлен первый принцип: инициализация. Делается так, чтобы оба распределения примерно совпадали, например, с помощью дословного перевода с использованием словаря. В пункте В проиллюстрирован второй принцип: языковое моделирование. Языковая модель обучается независимо для каждого стиля для определения структуры данных (изображено в виде непрерывной кривой); она действует как основанная на данных система для устранения шума / исправления предложений (проиллюстрировано пружиной, втягивающей предложение за пределы многообразия обратно). И, наконец, под пунктом Г изображен третий принцип: обратный перевод (см. 1.5.3). Начиная с наблюдаемого исходного предложения (заполненный красный круг), используется текущая модель для парафразы (пунктирная стрелка), что приводит к потенциально неправильному переносу стиля (синий крестик рядом с пустым кругом). Начиная с этого (обратного) перевода, используется модель для переноса стиля в обратную сторону (непрерывная стрелка) для восстановления предложения в оригинальном стиле. Несоответствие между восстановлением и исходным предложением является сигналом ошибки для обучения параметрам модели. Та же процедура применяется в обратном направлении.

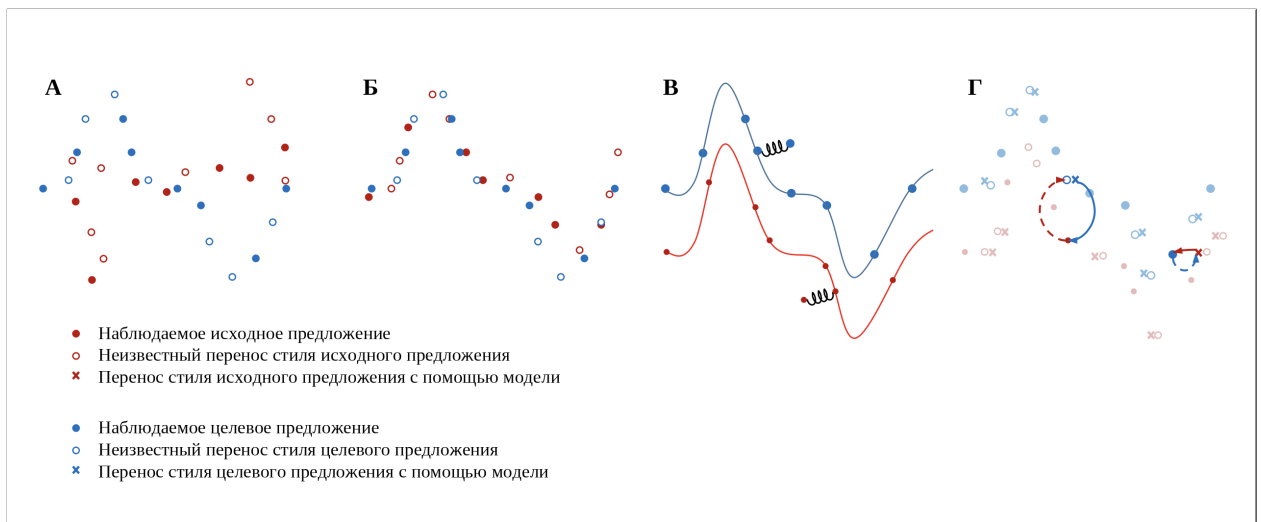


Рисунок 10 — Принципы алгоритмов "без учителя" для задачи переноса стиля текста

На этапе инициализации, вместо рассмотрения отдельных слов в предложении, используются byte-pair encoding токены (BPE-токены) [34]. Подобная токенизация даёт два больших преимущества: уменьшает размер словаря и избавляет от присутствия неизвестных UNK-токенов в перефразированном предложении. BPE-токены обучаются на совместном наборе данных обоих стилей, так как язык одинаковый и большая часть этих токенов будет совпадать. Помимо прочего, в таком случае отпадает необходимость получать словарь между словами разного стиля. Затем, обучаются word2vec эмбединги токенов [35], который затем будут использованы в языковой модели.

Этап языкового моделирования представлен автокодировщиком с шумоподавлением (denoising autoencoder, DAE) [36, 32]. Архитектурой автокодировщика является transformer [37]. Без каких-либо ограничений обычный автокодировщик очень быстро учится просто копировать каждое вводимое слово одно за другим. Такая модель также идеально копировала бы последовательности случайных слов, предполагая, что модель не выучивает какую-либо полезную структуру в данных. Чтобы решить эту проблему, во входные предложения добавляется шум [38] и применяется стратегия подавления шума, получая Denoising Autoencoder – автокодировщик с шумоподавлением [36]. Общая схема представлена на рисунке 11, а функция потерь задаётся как:

$$\mathcal{L}^{lm} = \mathbb{E}_{x \sim \mathcal{S}}[-\log P_{s \rightarrow s}(x|C(x))] + \mathbb{E}_{y \sim \mathcal{T}}[-\log P_{t \rightarrow t}(y|C(y))]$$

где C - функция зашумления, которая удаляет или меняет местами некоторые слова [31], а $P_{s \rightarrow s}$ и $P_{t \rightarrow t}$ - композиции энкодера и декодера, оперирующие в исходном или целевом пространстве соответственно.

Пространство предложений в исходном и целевом стиле обозначается через \mathcal{S} и \mathcal{T} соответственно, а языковые модели, обученные на исходных и целевых наборах данных, через P_s и P_t , соответственно. Через $P_{s \rightarrow t}$ и $P_{t \rightarrow s}$ обозначим модели парафразы из исходного стиля в целевой и наоборот. Пусть $u^*(y)$ это предложение в исходном стиле, полученное из $y \in \mathcal{T}$ такое, что $y^*(y) = \arg \max P_{t \rightarrow s}(u|y)$. Подобным образом пусть

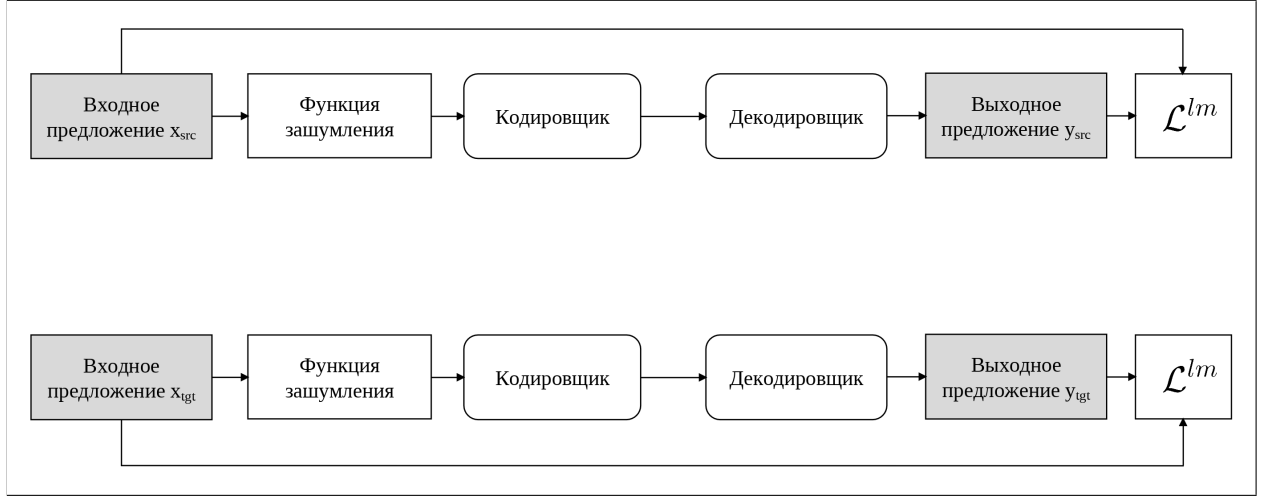


Рисунок 11 — Обучение denoising auto-encoder

$v^*(x) = \arg \max P_{s \rightarrow t}(v|x)$. Таким образом пара $((u^*(y), y)$ и $(x, v^*(x)))$ представляет автоматически перефразированные параллельные предложения, которые могут быть использованы для обучения модели, минимизируя следующую функцию потерь:

$$\mathcal{L}^{back} = \mathbb{E}_{y \sim \mathcal{T}}[-\log P_{s \rightarrow t}(y|u^*(y))] + \mathbb{E}_{x \sim \mathcal{S}}[-\log P_{t \rightarrow s}(x|v^*(x))]$$

Этим и выражается третий принцип. Общая схема представлена на рисунке 12.

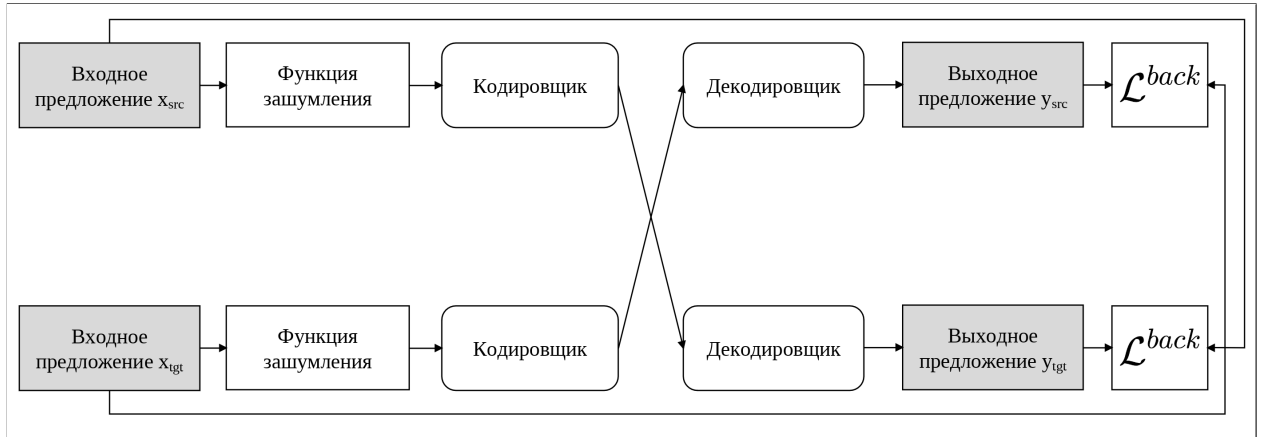


Рисунок 12 — Обучение с помощью back-translation

Ключевым параметром является общее использование энкодера между стилями. Что касается декодера, то авторы заявляют, что его использование оказывает крайне малое влияние [33] на качество обучения, однако в рамках данной работы это оказалось ключевым фактором. Если исполь-

зовать общий декодер, то модель крайне быстро сходится к тому, что копирует исходный текст. Поэтому в итоговой модели используется общий энкодер и два декодера на каждый из стилей.

Метрики качества указаны в таблице 3.1, а примеры генерации в приложении А.

3.4 Guided Generation

Большие языковые модели (large language models, LLMs), типа GPT [39, 40], способны достаточно хорошо изучить распределение своего обучающего набора данных, чтобы затем генерировать реалистичный текст. При обучении большой языковой модели используется огромный набор данных, собранный по всему интернету, в котором содержатся тексты различного содержания и различных стилей. Поэтому имеет смысл пытаться управлять генерацией языковой модели, чтобы результирующий текст имел желательную стилевую окраску.

В работах *GeDi: Generative Discriminator Guided Sequence Generation* [41] и *Text Detoxification using Large Pre-trained Neural Models* [42] предлагается управлять генерацией текста с помощью другой языковой модели, которая обучена на требуемом домене данных.

Предлагаемая модель GeDi состоит из двух компонент: генеративная модель на основе GPT-2 и дискриминативная модель, тоже на основе GPT-2, но обученная с дополнительными метками стиля на уровне предложений. Это заставляет модель дискриминатора выучивать распределения слов, обусловленные конкретным стилем. На каждом шаге генерации, распределение токенов, полученное из основной модели P_{LM} редактируется с помощью модели дискриминатора P_D и правила Байеса:

$$P(x_t|x_{<t},c) \propto P_{LM}(x_t|x_{<t})P_D(c|x_t,x_{<t})$$

где x_t это текущий токен, $x_{<t}$ уже сгенерированный текст, а c это желаемый стилистический атрибут (один из C классов). Первый член генерируется основной языковой моделью P_{LM} , а второй вычисляется с помощью правила Байеса и дополнительной условной языковой модели P_{CC} . Таким образом,

токены, которые более вероятно окажутся в тексте желаемым стилем, получают более высокую вероятность:

$$P_D(c|x_t, x_{<t}) \propto P(c)P_{CC}(x, x_{<t}|c)$$

Принцип работы проиллюстрирован на рисунке 13.

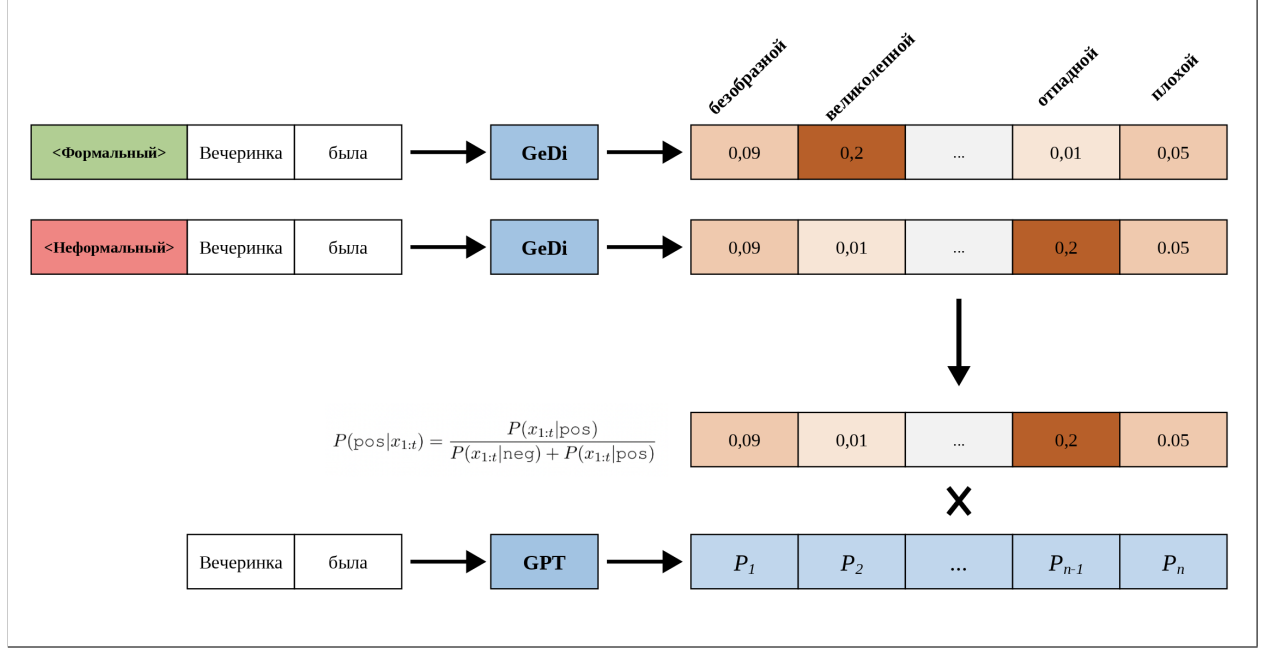


Рисунок 13 — Модель GeDi

Чтобы сохранить содержание предложения в работе [42] предлагается заменить основную языковую модель на модель, обученную на генерацию парафраза. Пусть x это исходное предложение, T длина сгенерированного текста y , а c это желаемый стиль, то предлагаемая модель моделирует следующую вероятность:

$$P(y_t|y_{<t}, x, c) \propto P_{LM}(y_t|y_{<t}, x)P(c|y_t, y_{<t}, x) \approx P_{LM}(y_t|y_{<t}, x)P_D(c|y_t, y_{<t})$$

Последнее это аппроксимация, потому что вероятность класса должна быть обусловлена как x , так и y . Однако это приближение, хотя и не является полностью оправданным, позволяет отделить модель парафраза (которая требует параллельного корпуса для обучения) от модели стиля (которая требует только текстов с метками стиля, не обязательно параллельных). Общий принцип работы изображен на рисунке 14

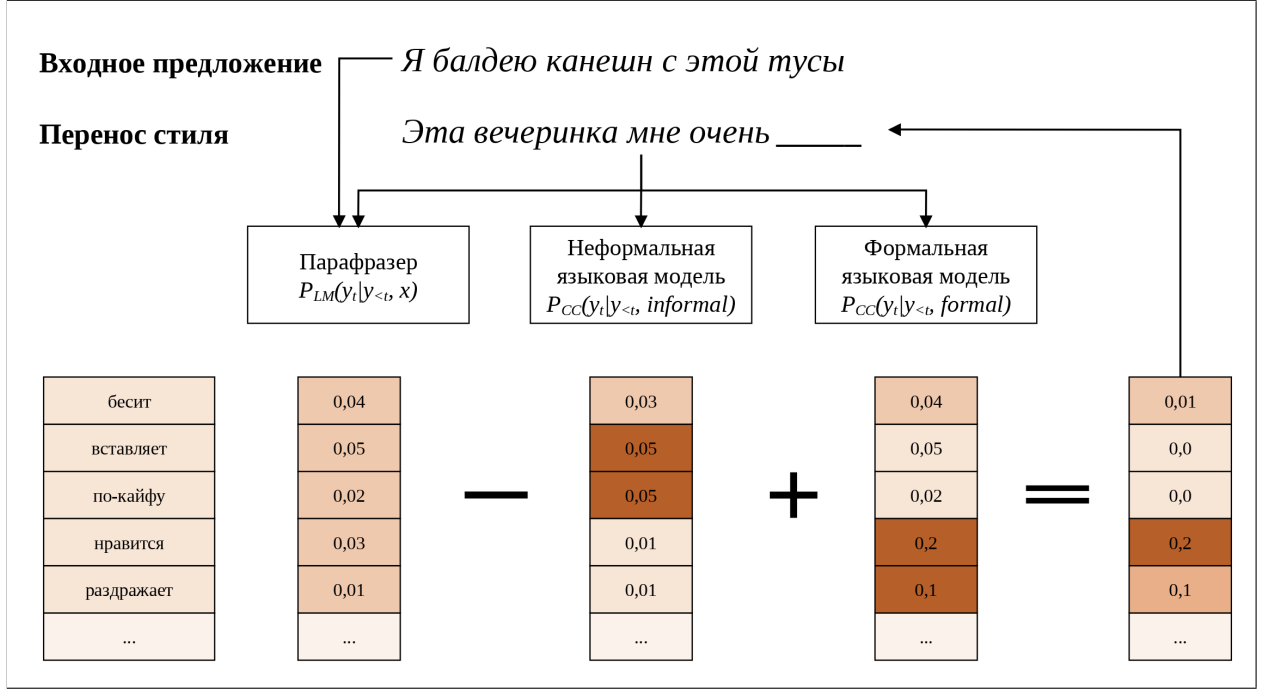


Рисунок 14 — Использование модели парафразера в GeDi

Функция потерь $\mathcal{L}_{ParaGeDi}$ состоит из линейной комбинации двух других функций потерь: генеративной \mathcal{L}_G , используемого в обучении языковой модели, и дискриминативной \mathcal{L}_D , которая отдаляет разные классы друг от друга в латентном пространстве.

$$\mathcal{L}_G = -\frac{1}{N} \sum_{i=1}^N \frac{1}{T_i} \sum_{t=1}^{T_i} \log P(y_t^{(i)} | y_{<t}^{(i)}, c^{(i)})$$

$$\mathcal{L}_D = -\frac{1}{N} \sum_{i=1}^N \log P(c^{(i)} | y_{1:T_i}^{(i)})$$

$$\mathcal{L}_{ParaGeDi} = \lambda \mathcal{L}_D + (1 - \lambda) \mathcal{L}_G$$

В качестве условной языковой модели была использована `ai-forever/rugpt3large_based_on_gpt2`, а в качестве модели парафразы был взят предобученный `cointegrated/rut5-base-paraphraser`.

Метрики качества итогового алгоритма представлены в таблице 3.1, а примеры генерации в приложении А.

Заметим, что основанные на референсах n-граммные метрики значительно ухудшились, но в то же время можно наблюдать вполне удовлетворительные парафразы. Здесь стоит сделать вывод и подтвердить другие

недавние исследования [4, 8, 43] о том, что несмотря на наличие параллельного набора данных данные метрики для этой задачи являются не самыми адекватными. Ведь каждый конкретный стиль может быть выражен разными конструкциями и нет однозначного сопоставления.

3.5 Parameter-efficient fine-tuning

Параметро-эффективные методы (Parameter-Efficient Fine-Tuning, PEFT) зарекомендовали себя как очень простой и в то же время крайне результативный подход к решению большинства задач.

Общая идея всех методов сводится к:

- а) использованию больших предобученных языковых моделей (Large Language Models, LLMs);
- б) заморозке всех её параметров;
- в) добавлению небольшого (могут быть десятые доли одного процента от всех параметров) количества дообучаемых параметров;
- г) обучению этих параметров на небольшом наборе данных.

Такой подход даёт следующие преимущества:

- если, при наличии набора данных низкого качества, во время классического дообучения (fine-tuning) модель могла ухудшить качество на некоторых задачах, то в этом семействе методов подобного не наблюдается;
- значительно снижается необходимое количество размеченных данных — от нескольких сотен до нескольких тысяч примеров;
- удобство использования в итоговой системе, потому что достаточно иметь одну большую модель и сколь угодно много адаптеров к ней, которые можно быстро переключать между собой по мере необходимости;
- легко дообучать на поступающих новых данных.

3.5.1 LoRA

Low-Rank Adaptation of Large Language Models (LoRA) – метод дообучения, основанный на заморозке весов предобученной модели и добавлении

обучаемых матриц ранговой декомпозиции в каждый слой архитектуры модели, значительно сокращая количество обучаемых параметров [30].

Пусть имеется предобученная авторегрессивная языковая модель $P_\Phi(y|x)$, параметризованная Φ . Каждая задача представлена обучающим набором данных пар контекст-ответ: $\mathcal{Z} = \{(x_i, y_i)\}_{i=1, \dots, N}$, где оба x_i и y_i представляют собой последовательность токенов. Во время полноценного дообучения (fine-tuning) модель инициализирована предобученными весами Φ_0 и обновляется как $\Phi_0 + \Delta\Phi$, многократно следуя градиенту для решения максимизационной задачи условного языкового моделирования:

$$\max_{\Phi} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(P_\Phi(y_t|x, y_{<t}))$$

Основным недостатком является то, что для каждой задачи обучается различный набор параметров $\Delta\Phi$, размерность которого $|\Delta\Phi|$ равна $|\Phi_0|$. Соответственно, с увеличением размера предобученной модели, хранение и разворачивание множества реплик предобученной модели становится всё более сложной задачей.

В данном подходе применяется более эффективный с точки зрения параметров подход, при котором приращение параметра $\Delta\Phi = \Delta\Phi(\Theta)$ для конкретной задачи дополнительно кодируется набором параметров Θ гораздо меньшего размера $|\Theta| \ll |\Phi_0|$. Задача нахождения $\Delta\Phi$ превращается в задачу оптимизации по Θ :

$$\max_{\Theta} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(p_{\Phi_0 + \Delta\Phi(\Theta)}(y_t|x, y_{<t}))$$

Нейронная сеть состоит из множества полносвязных слоёв, которые выполняют матричное перемножение. Матрицы весов в этих слоях обычно являются полноранговыми. Исследования показывают, что при адаптации к специфической задаче предобученные подмодели имеют маленькую „внутреннюю размерность“ и могут эффективно обучаться, несмотря на случайную проекцию в меньшее подпространство [44]. Из данного наблюдения вытекает гипотеза, что обновления веса моделей тоже имеют небольшую

„внутреннюю размерность“. Для предобученной матрицы весов $W_0 \in \mathbb{R}^{d \times k}$ ограничивается её обновления, представляя его в виде разложения низкого ранга $W_0 + \Delta W = W_0 + BA$, где $B \in \mathbb{R}^{d \times k}$, $A \in \mathbb{R}^{r \times k}$ и ранг $r \ll \min(d, k)$. Во время обучения матрица W_0 заморожена и не получает обновлений от градиента, в то время как A и B являются обучаемыми параметрами. Оба W_0 и $\Delta W = BA$ умножаются на одинаковый вектор, а их соответствующие выходные вектора по-координатно суммируются. Для $h = W_0x$ модифицированный прямой проход:

$$h = W_0x + \Delta Wx = W_0x + BAx$$

Данное выражение изображено на рисунке 15. Для A используется случайная Гауссова инициализация, а B инициализируется нулями, так что $\Delta W = BA$ равна нулю в начале обучения.

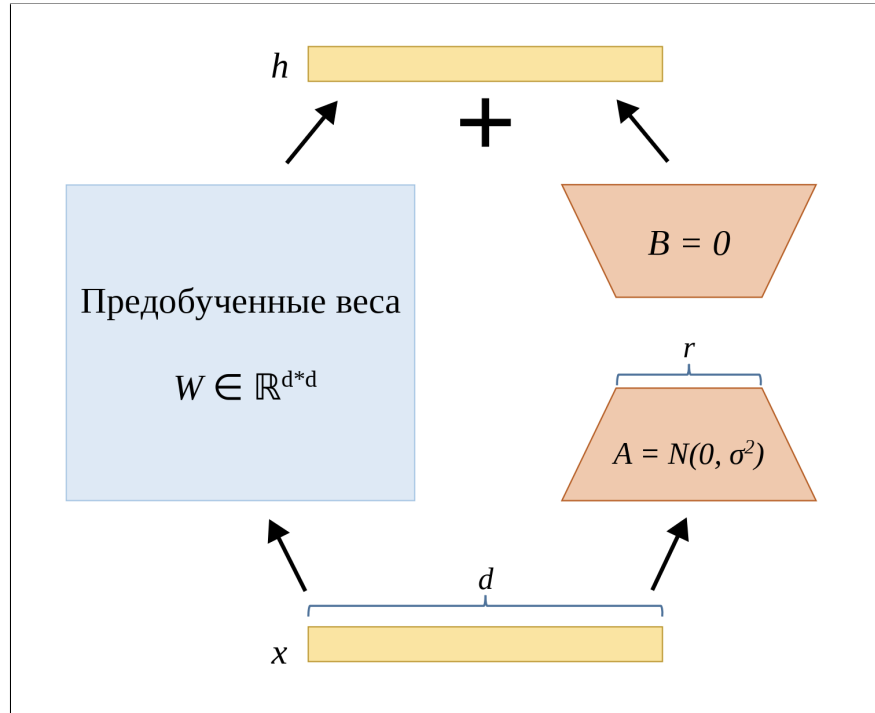


Рисунок 15 — Параметризация весов в алгоритме LoRA. Обновляются только матрицы A и B

В работе с помощью данного алгоритма была дообучена модель `cointegrated/rut5-base-paraphraser` на собранном наборе данных. Метрики качества указаны в таблице 3.1, а примеры генерации в в приложении А.

3.5.2 P-Tuning

Данный метод основан на автоматическом поиске затравок (prompts) в непрерывном пространстве [45].

Пусть дана предобученная языковая модель \mathcal{M} . Последовательность дискретных входных токенов $x_{1:n} = \{x_0, x_1, \dots, x_n\}$ будет сопоставлена с входными эмбедами $\{\mathbf{e}(x_0), \mathbf{e}(x_1), \dots, \mathbf{e}(x_n)\}$ с помощью предварительно обученного слоя эмбедингов $\mathbf{e} \in \mathcal{M}$. В конкретном случае, зависящем от контекста \mathbf{x} , часто используются выходные эмбединги набора целевых токенов \mathbf{y} для последующей обработки.

Функция затравки (промпта) \mathbf{p} - организовать контекст \mathbf{x} , результат \mathbf{y} и саму себя в шаблон T . Для задачи переноса стиля текста шаблоном может быть «Перепиши предложение „ \mathbf{x} “ в неформальном стиле: „ \mathbf{y} “». Здесь «Перепиши предложение ... в неформальном стиле: ...» это затравка, \mathbf{x} это контекст, а \mathbf{y} это результат.

Пусть \mathcal{V} это словарный запас языковой модели \mathcal{M} , а $[P_i]$ является i -ым токеном шаблона T .

Имея шаблон $T = \{[P_{0:i}], x, [P_{i+1:m}], y\}$, дискретная затравка удовлетворяет условию $[P_i] \in \mathcal{V}$ и превращает T в

$$\{\mathbf{e}([P_{0:i}]), \mathbf{e}(\mathbf{x}), \mathbf{e}([P_{i+1:m}]), \mathbf{e}(\mathbf{y})\}$$

В отличие от этого, P-tuning воспринимает $[P_i]$ как псевдо-токены и превращает шаблон в

$$\{h_0, \dots, h_i, \mathbf{e}(\mathbf{x}), h_{i_1}, \dots, h_m, \mathbf{e}(\mathbf{x})\}$$

где $h_i (0 \leq i < m)$ это обучаемые эмбединги. Это позволяет находить лучшие непрерывные затравки вне словарного запаса, которым оперирует языковая модель. Сравнение дискретного поиска и P-tuning приллюстрировано на рисунке 16.

Имея функцию потерь \mathcal{L} можно дифференциально оптимизировать непрерывные затравки $h_i (0 \leq i < m)$ как

$$\hat{h}_{0:m} = \arg \min_h \mathcal{L}(\mathcal{M}(\mathbf{x}, \mathbf{y}))$$

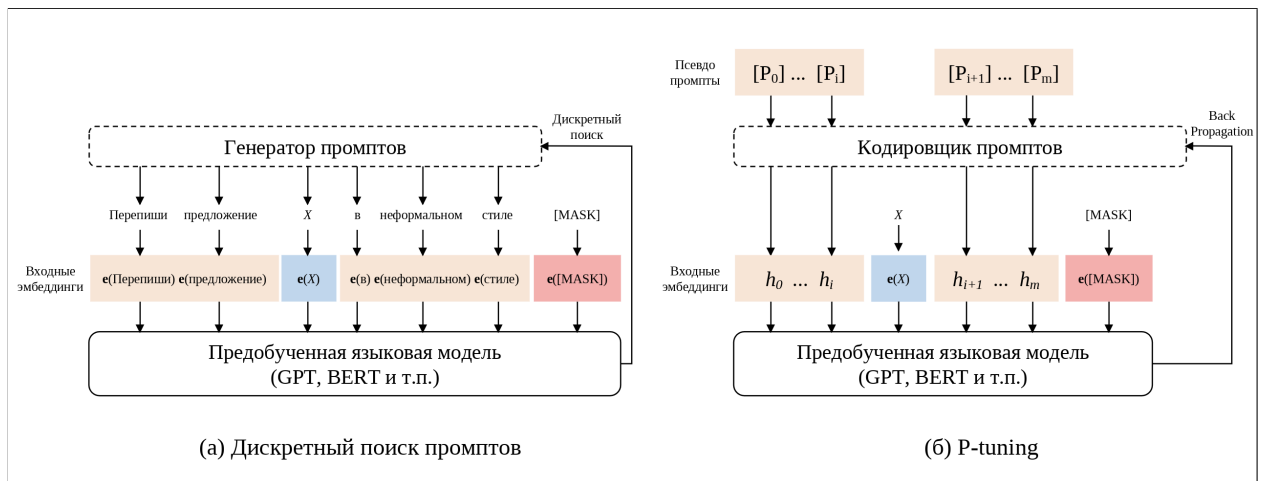


Рисунок 16 — P-tuning

В работе с помощью данного алгоритма была дообучена модель `ai-forever/rugpt3small_based_on_gpt2` на собранном наборе данных. Метрики качества указаны в таблице 3.1, а примеры генерации в в приложении А.

3.6 Few-Shot LLM

Появление таких больших генеративных языковых моделей, как ChatGPT и GPT-4 [40, 46] показало, что после превышения определённого числа параметров модель становится достаточно большой, чтобы решать большинство задач во few-shot или даже zero-shot режиме.

В данной работе так же был исследован вопрос, как хорошо справится подобная модель с задачей переноса стиля текста. Для этого была выбрана модель GPT-4 от OpenAI. Была подобрана затравка с несколькими примерами изменения формальности текста и модели было дано задание по этому подобию изменить формальность других предложений. Пример затравки и генераций представлен в приложении А. Можно сделать вывод, что GPT-4 полностью справился с поставленной ему задачей.

3.7 Результаты

Результаты метрик по всем проведенным экспериментам представлены в таблице 3.1, а примеры генераций в приложении А.

Можно сделать вывод, что лучше всего показали себя большие языковые модели, способные без какого-либо дообучения на конкретную задачу показывать результат лучше, чем целенаправленно обученные на задачу модели. Языковые модели прошлого поколения (GPT-3, T-5 и пр.) с помощью параметро-эффективных методов обучения могут показать хороший результат, особенно в случае отсутствия большого набора данных. Более старые алгоритмы показывают качество генераций намного хуже.

Обучение без учителя и Guided Generation показали лучшие метрики SacreBLEU и METEOR. Использование предобученных языковых моделей в сочетании с параметро-эффективными методами дообучения и использование собранного параллельного набора данных дало наилучший результат с точки зрения комбинации метрик Style Score и Semantic Score. Это подтверждает другие исследования [4, 8], что использование метрик, основанных на n-граммах и сравнении с референсами, зачастую не даёт адекватной оценки качества алгоритма.

Таблица 3.1 — Метрики качества исследованных подходов

Метод	SacreBLEU	METEOR	StyleScore	SemanticScore
	wiki → lurk / lurk → wiki			
Unsupervised DAE	40,5/38,8	0,6/0,59	0,22/0,6	0,82/0,78
Guided Generation	12,7/6,7	0,35/0,27	0,13/0,72	0,81/0,75
LoRA(ruT5-paraphraser)	3,04/3,35	0,25/0,26	0,3/0,75	0,79/0,8
P-Tuning (GPT)	3,16/4,56	0,23/0,23	0,56/0,92	0,73/0,78

Также обратим внимание, что с точки зрения StyleScore задача трансфера стиля из формального вики в неформальный лурк является намного более сложной задачей, с которой во всех рассмотренных методах наблюдаются трудности. Как было упомянуто в параграфе 1.4 перефразирование в неформальный стиль является более трудной задачей даже для человека. Стилистическая специфика лурка более сложна и модели труднее её выучить, нежели формальный стиль Википедии.

4 Развитие работы

4.1 Продолжение создания набора данных

Приняв во внимания аспекты из главы 2, а так же усовершенствование алгоритмов классификации, имеет смысл продолжить создание набора данных на новом качественном уровне. Усовершенствованные алгоритмы позволят более качественно фильтровать изначальный корпус и выдавать ассессорам более качественные предложения. Тем временем, улучшая взаимодействие с ассессорами, можно добиться более высокого итогового качества.

4.2 Улучшение метрик качества

Как было упомянуто в главе 3, данная работа подтвердила другие недавние исследования, указывавшие на проблему соответствия автоматических метрик человеческим оценкам в общем, и в использовании референсных метрик в частности. Существующие ограничения метрик оценки качества мотивируют к изучению новых способов оценки качества работы моделей по переносу стиля текста.

4.3 Улучшение алгоритмов

Исследование показало, что сбор набора данных хоть и посильная, но очень трудозатратная активность. В связи с этим дальнейшее изучение и улучшение алгоритмов, не требующих параллельного набора данных, является перспективным направлением. Особенно в рамках использования больших языковых моделей, показывающих впечатляющий результат, без дообучения на конкретную задачу. Несмотря на это, большие языковые модели являются чрезмерно громоздкими и работа по уменьшению их размера для использования в задаче переноса стиля текста является обещающим направлением для исследования.

Заключение

Был собран набор данных для переноса формальности на русском языке объёмом 18 тысяч параллельных пар формальное-неформальное предложение. Данного набора данных достаточно, для оценки качества моделей, а так же их обучению в случае использования параметро-эффективных методов обучения с современными языковыми моделями.

Можно сделать вывод, что лучше всего в решении поставленной задачи показали себя большие языковые модели, способные без какого-либо дообучения на конкретную задачу показывать результат лучше, чем целенаправленно обученные на задачу модели. Языковые модели прошлого поколения (GPT-2, GPT-3, T-5 и пр.) с помощью параметро-эффективных методов обучения могут показать хороший результат, особенно в случае отсутствия большого набора данных. Более поздние алгоритмы показывают качество генераций намного хуже.

Обучение без учителя и Guided Generation показали лучшие метрики SacreBLEU и METEOR. Использование предобученных языковых моделей в сочетании с параметро-эффективными методами дообучения и использование собранного параллельного набора данных дало наилучший результат с точки зрения комбинации метрик Style Score и Semantic Score. Это подтверждает другие исследования, что использование метрик, основанных на n-граммах и сравнении с референсами, зачастую не даёт адекватной оценки качества алгоритма. Данное наблюдение поднимает вопрос о необходимости переосмысления подходов к оценке качества алгоритмов переноса стиля.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *McDonald, David D.* A Computational Theory of Prose Style for Natural Language Generation / David D. McDonald, James D. Pustejovsky // Second Conference of the European Chapter of the Association for Computational Linguistics. — Geneva, Switzerland: Association for Computational Linguistics, 1985. — . <https://aclanthology.org/E85-1027>.
2. Deep Learning for Text Style Transfer: A Survey / Di Jin, Zhi-jing Jin, Zhiting Hu et al. // *CoRR*. — 2020. — Vol. abs/2011.00416. <https://arxiv.org/abs/2011.00416>.
3. Evaluating the Evaluation Metrics for Style Transfer: A Case Study in Multilingual Formality Transfer / Eleftheria Briakou, Sweta Agrawal, Joel R. Tetreault, Marine Carpuat // *CoRR*. — 2021. — Vol. abs/2110.10668. <https://arxiv.org/abs/2110.10668>.
4. Delete, Retrieve, Generate: A Simple Approach to Sentiment and Style Transfer / Juncen Li, Robin Jia, He He, Percy Liang // *CoRR*. — 2018. — Vol. abs/1804.06437. <http://arxiv.org/abs/1804.06437>.
5. BLEU: a Method for Automatic Evaluation of Machine Translation / Kishore Papineni, Salim Roukos, Todd Ward, Wei Jing Zhu. — 2002. — 10.
6. *Post, Matt.* A Call for Clarity in Reporting BLEU Scores / Matt Post // Proceedings of the Third Conference on Machine Translation: Research Papers. — Belgium, Brussels: Association for Computational Linguistics, 2018. — . — Pp. 186–191. <https://www.aclweb.org/anthology/W18-6319>.
7. *Hu, Zhiqiang.* Text Style Transfer: A Review and Experiment Evaluation / Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal // *CoRR*. — 2020. — Vol. abs/2010.12742. <https://arxiv.org/abs/2010.12742>.
8. Evaluating Style Transfer for Text / Remi Mir, Bjarke Felbo, Nick Obradovich, Iyad Rahwan // *CoRR*. — 2019. — Vol. abs/1904.02295. <http://arxiv.org/abs/1904.02295>.
9. *Lavie, Alon.* METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments / Alon Lavie, Abhaya Agarwal. — 2007. — 07. — Pp. 228–231.

10. *Lin, Chin-Yew*. ROUGE: A Package for Automatic Evaluation of Summaries / Chin-Yew Lin // Text Summarization Branches Out. — Barcelona, Spain: Association for Computational Linguistics, 2004. — . — Pp. 74–81. <https://aclanthology.org/W04-1013>.
11. Style Transfer in Text: Exploration and Evaluation / Zhenxin Fu, Xiaoye Tan, Nanyun Peng et al. // *CoRR*. — 2017. — Vol. abs/1711.06861. <http://arxiv.org/abs/1711.06861>.
12. BERTScore: Evaluating Text Generation with BERT / Tianyi Zhang, Varsha Kishore, Felix Wu et al. // *CoRR*. — 2019. — Vol. abs/1904.09675. <http://arxiv.org/abs/1904.09675>.
13. Unsupervised Text Style Transfer using Language Models as Discriminators / Zichao Yang, Zhiting Hu, Chris Dyer et al. // *CoRR*. — 2018. — Vol. abs/1805.11749. <http://arxiv.org/abs/1805.11749>.
14. *Pang, Richard Yuanzhe*. Learning Criteria and Evaluation Metrics for Textual Transfer between Non-Parallel Corpora / Richard Yuanzhe Pang, Kevin Gimpel // *CoRR*. — 2018. — Vol. abs/1810.11878. <http://arxiv.org/abs/1810.11878>.
15. ReproGen: Proposal for a Shared Task on Reproducibility of Human Evaluations in NLG / Anya Belz, Shubham Agarwal, Anastasia Shimorina, Ehud Reiter // Proceedings of the 13th International Conference on Natural Language Generation. — Dublin, Ireland: Association for Computational Linguistics, 2020. — . — Pp. 232–236. <https://aclanthology.org/2020.inlg-1.29>.
16. *Rao, Sudha*. Dear Sir or Madam, May I introduce the YAFC Corpus: Corpus, Benchmarks and Metrics for Formality Style Transfer / Sudha Rao, Joel R. Tetreault // *CoRR*. — 2018. — Vol. abs/1803.06535. <http://arxiv.org/abs/1803.06535>.
17. XFORMAL: A Benchmark for Multilingual Formality Style Transfer / Eleftheria Briakou, Di Lu, Ke Zhang, Joel R. Tetreault // *CoRR*. — 2021. — Vol. abs/2104.04108. <https://arxiv.org/abs/2104.04108>.
18. *Sennrich, Rico*. Improving Neural Machine Translation Models with Monolingual Data / Rico Sennrich, Barry Haddow, Alexandra Birch // *CoRR*. — 2015. — Vol. abs/1511.06709. <http://arxiv.org/abs/1511.06709>.

06709.

19. Unsupervised Text Style Transfer via Iterative Matching and Translation / Zhijing Jin, Di Jin, Jonas Mueller et al. // *CoRR*. — 2019. — Vol. abs/1901.11333. <http://arxiv.org/abs/1901.11333>.

20. Iterative Back-Translation for Neural Machine Translation / Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, Trevor Cohn // Proceedings of the 2nd Workshop on Neural Machine Translation and Generation. — Melbourne, Australia: Association for Computational Linguistics, 2018. — . — Pp. 18–24. <https://aclanthology.org/W18-2703>.

21. Controllable Text Generation / Zhiting Hu, Zichao Yang, Xiaodan Liang et al. // *CoRR*. — 2017. — Vol. abs/1703.00955. <http://arxiv.org/abs/1703.00955>.

22. *Kingma, Diederik P.* Auto-Encoding Variational Bayes. — 2022.

23. *Mueller, Jonas.* Sequence to Better Sequence: Continuous Revision of Combinatorial Structures / Jonas Mueller, David Gifford, Tommi Jaakkola // Proceedings of the 34th International Conference on Machine Learning / Ed. by Doina Precup, Yee Whye Teh. — Vol. 70 of *Proceedings of Machine Learning Research*. — PMLR, 2017. — 06–11 Aug. — Pp. 2536–2544. <https://proceedings.mlr.press/v70/mueller17a.html>.

24. *Williams, Ronald J.* Simple statistical gradient-following algorithms for connectionist reinforcement learning / Ronald J. Williams // *Machine Learning*. — 1992. — Vol. 8. — Pp. 229–256.

25. A Dual Reinforcement Learning Framework for Unsupervised Text Style Transfer / Fuli Luo, Peng Li, Jie Zhou et al. // *CoRR*. — 2019. — Vol. abs/1905.10060. <http://arxiv.org/abs/1905.10060>.

26. *Sharov, S.* Functional text dimensions for annotation of web corpora / S Sharov // *Corpora*. — 2016. — 07. — Pp. 65–95. <https://eprints.whiterose.ac.uk/102914/1/2018-ftd.pdf>.

27. Crowdsourcing of Parallel Corpora: the Case of Style Transfer for Detoxification / Daryna Dementieva, Sergey Ustyantsev, David Dale et al. // Proceedings of the 2nd Crowd Science Workshop: Trust, Ethics, and Excellence in Crowdsourced Data Management at Scale co-located with 47th International Conference on Very Large Data Bases (VLDB

2021 (<https://vldb.org/2021/>). — Copenhagen, Denmark: CEUR Workshop Proceedings, 2021. — Pp. 35–49. <http://ceur-ws.org/Vol-2932/paper2.pdf>.

28. PyTorch: An Imperative Style, High-Performance Deep Learning Library / Adam Paszke, Sam Gross, Francisco Massa et al. // *CoRR*. — 2019. — Vol. abs/1912.01703. <http://arxiv.org/abs/1912.01703>.

29. HuggingFace’s Transformers: State-of-the-art Natural Language Processing / Thomas Wolf, Lysandre Debut, Victor Sanh et al. // *CoRR*. — 2019. — Vol. abs/1910.03771. <http://arxiv.org/abs/1910.03771>.

30. LoRA: Low-Rank Adaptation of Large Language Models / Edward J. Hu, Yelong Shen, Phillip Wallis et al. // *CoRR*. — 2021. — Vol. abs/2106.09685. <https://arxiv.org/abs/2106.09685>.

31. *Lample, Guillaume*. Unsupervised Machine Translation Using Monolingual Corpora Only / Guillaume Lample, Ludovic Denoyer, Marc’Aurelio Ranzato // *CoRR*. — 2017. — Vol. abs/1711.00043. <http://arxiv.org/abs/1711.00043>.

32. Phrase-Based & Neural Unsupervised Machine Translation / Guillaume Lample, Myle Ott, Alexis Conneau et al. // *CoRR*. — 2018. — Vol. abs/1804.07755. <http://arxiv.org/abs/1804.07755>.

33. Multiple-Attribute Text Style Transfer / Sandeep Subramanian, Guillaume Lample, Eric Michael Smith et al. // *CoRR*. — 2018. — Vol. abs/1811.00552. <http://arxiv.org/abs/1811.00552>.

34. *Sennrich, Rico*. Neural Machine Translation of Rare Words with Subword Units / Rico Sennrich, Barry Haddow, Alexandra Birch // *CoRR*. — 2015. — Vol. abs/1508.07909. <http://arxiv.org/abs/1508.07909>.

35. Distributed Representations of Words and Phrases and their Compositionality / Tomás Mikolov, Ilya Sutskever, Kai Chen et al. // *CoRR*. — 2013. — Vol. abs/1310.4546. <http://arxiv.org/abs/1310.4546>.

36. Extracting and composing robust features with denoising autoencoders / Pascal Vincent, Hugo Larochelle, Y. Bengio, Pierre-Antoine Manzagol. — 2008. — 01. — Pp. 1096–1103.

37. Attention Is All You Need / Ashish Vaswani, Noam Shazeer, Niki Parmar et al. // *CoRR*. — 2017. — Vol. abs/1706.03762. <http://arxiv.org/abs/1706.03762>.

[//arxiv.org/abs/1706.03762](https://arxiv.org/abs/1706.03762).

38. *Hill, Felix*. Learning Distributed Representations of Sentences from Unlabelled Data / Felix Hill, Kyunghyun Cho, Anna Korhonen // *CoRR*. — 2016. — Vol. abs/1602.03483. <http://arxiv.org/abs/1602.03483>.

39. Language Models are Unsupervised Multitask Learners / Alec Radford, Jeff Wu, Rewon Child et al. — 2019.

40. Language Models are Few-Shot Learners / Tom B. Brown, Benjamin Mann, Nick Ryder et al. // *CoRR*. — 2020. — Vol. abs/2005.14165. <https://arxiv.org/abs/2005.14165>.

41. GeDi: Generative Discriminator Guided Sequence Generation / Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann et al. // *CoRR*. — 2020. — Vol. abs/2009.06367. <https://arxiv.org/abs/2009.06367>.

42. Text Detoxification using Large Pre-trained Neural Models / David Dale, Anton Voronov, Daryna Dementieva et al. // *CoRR*. — 2021. — Vol. abs/2109.08914. <https://arxiv.org/abs/2109.08914>.

43. *Shen, Lingfeng*. On the Evaluation Metrics for Paraphrase Generation. — 2022.

44. *Aghajanyan, Armen*. Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning / Armen Aghajanyan, Luke Zettlemoyer, Sonal Gupta // *CoRR*. — 2020. — Vol. abs/2012.13255. <https://arxiv.org/abs/2012.13255>.

45. GPT Understands, Too / Xiao Liu, Yanan Zheng, Zhengxiao Du et al. // *CoRR*. — 2021. — Vol. abs/2103.10385. <https://arxiv.org/abs/2103.10385>.

46. *OpenAI*. GPT-4 Technical Report. — 2023.

Благодарности

Хочу выразить благодарность людям, помогавшим в выполнении работы:

- Марии Пономарёвой – за научное руководство и критику;
- Александре Ивойловой – за помощь в создании набора данных;
- Студентам РГГУ – за разметку данных.

Приложение А Примеры генераций

Lurk: В разделе «Интеллектуальная собственность» периодически раздаются вопли копирастов.

Wiki: В разделе «Интеллектуальная собственность» периодически раздаются вопли сторонников охраны авторских прав.

Lurk → Wiki: В разделе «Социальная собственность» периодически раздаются протесты властей

Wiki → Lurk: В разделе *Социальная собственность* периодически раздаются вопли сторонников охраны авторских прав

Lurk: После многочисленных высказываний об ущербности интернет-юзеров Михалков напилит видеоблог.

Wiki: После многочисленных высказываний о неполноценности интернет-пользователей Михалков начал вести видеоблог.

Lurk → Wiki: После многочисленных высказываний об недоности интернет-пользователей Иванов напилит видеоблог.

Wiki → Lurk: После многочисленных высказываний о неполноценности недопользователей Мавроди начал вести видеоблог.

Lurk: Алсо, пацанчики с шиномонтажа обширно ловят лулзы со срочников.

Wiki: Также, парни с шиномонтажа обширно смеются над срочниками.

Lurk → Wiki: Также, рассказчики с шиномонтажа обширно ловят значительные запасы со срочников.

Wiki → Lurk: Алсо, парни с шиномонтажа обширно смеются над срочниками.

Рисунок А.1 — Примеры генераций метода DAE

Lurk: В разделе «Интеллектуальная собственность» периодически раздаются вопли копирастов.

Wiki: В разделе «Интеллектуальная собственность» периодически раздаются вопли сторонников охраны авторских прав.

Lurk → Wiki: В рубрике «Интеллектуальная собственность» периодические обращения сторонников защиты авторских прав

Wiki → Lurk: В разделе «Интеллектуальная собственность» нередко раздаются вопли копирайтеров.

Lurk: После многочисленных высказываний об ущербности интернет-юзеров Михалков напилел видеоблог.

Wiki: После многочисленных высказываний о неполноценности интернет-пользователей Михалков начал вести видеоблог.

Lurk → Wiki: В связи с многочисленными сообщениями о некорректности использования Интернета, Михаилков приступил к созданию видео-блога.

Wiki → Lurk: После нескольких высказываний Михалкова о недоверии к интернет-пользователям, он начал создавать видеоблоги.

Lurk: Недавно МВД выпустили свой комикс дабы научить личинок и школоту порядку на улице.

Wiki: Недавно МВД выпустили свой комикс в целях научить детей и школьников порядку на улице.

Lurk → Wiki: Недавно МВД сделали комиксы с целью научить деток порядку.

Wiki → Lurk: «МВД» же недавно выпустили комиксы с целью научить школьников к порядку здешней школе.

Рисунок А.2 — Примеры генераций метода Guided Generation

Lurk: В разделе «Интеллектуальная собственность» периодически раздаются вопли копирастов.

Wiki: В разделе «Интеллектуальная собственность» периодически раздаются вопли сторонников охраны авторских прав.

Lurk → Wiki: Например, в разделе «Информация о пользователе» периодически появляются призывы к написанию статей.

Wiki → Lurk: Примерно раз в месяц раздаётся вопиющая *** о том, что авторские права на произведения принадлежат исключительно правообладателям.

Lurk: После многочисленных высказываний об ущербности интернет-юзеров Михалков напиллил видеоблог.

Wiki: После многочисленных высказываний о неполноценности интернет-пользователей Михалков начал вести видеоблог.

Lurk → Wiki: После нескольких высказаний о недоумении интернет-бользователей, Михалкова начала ведение видеоблога.

Wiki → Lurk: В связи с многочисленными высказываниями об ухудшении интернет-юзеров Михалкова написала видеообзор.

Lurk: Пост внезапно породил огромное количество обличительных комментариев и припоминаний автору его пророчеств и высказываний, спустя сутки сильно стремящиеся к 9000.

Wiki: Как ни странно, запись спровоцировала огромное количество обличительных комментариев и припоминаний автору его пророчеств и высказываний, спустя сутки сильно стремящихся к бесконечности.

Lurk → Wiki: Пост неожиданно породил большое количество оскорбительных комментариев и воспоминаний автору его предсказаний и высказываниях, через сутки очень стремящимся к 8000.

Wiki → Lurk: Скажем так, запись вызвала колоссальное количество лулзов и припиваний автору его предсказаний и высказываниям, через сутки изрядно стремящимся к бесконечности.

Рисунок А.3 — Примеры генераций метода P-Tunning



Твоя задача перефразировать текст из стиля Лурк в формальный стиль. Далее приведены 3 примеров перефразирования:



Лурк:

Тутта Ларсен — на самом деле никакая не Тутта, а вовсе даже Таня, долбанутая на всю верхнюю голову самка, бывшая лицом российского MTV, а на «Маяке» — главной разносчицей ПГМ.

Перефраз:

Тутта Ларсен — настоящее имя — Татьяна, отличается своей неординарностью, ранее являлась лицом российского телеканала MTV и ведущей радио «Маяк».

Лурк:

Буквально к концу 2013-го года Чумная Наташка решила порадовать гробофагов — снять документальный фильм с претендованием на эпичность: «Гражданская оборона. Начало».

Перефраз:

Точно к концу 2013-го года Чумная Наташка решила порадовать фанатов Гражданской обороны — снять документальный фильм с претендованием на исключительность: «Гражданская оборона. Начало».

Лурк:

Доставляют только хендмейдные игрушки некоторых участников, а также вполне винrarные «слизнячки» юзера, которые уже совсем не злы, а даже наоборот.

Перефраз:

Интерес представляют только игрушки ручной работы от некоторых участников, а также достаточно хорошо получившиеся «слизнячки» пользователя, которые, в отличие от оригинала, добрые.

Далее я буду посылать тебе предложения в стиле Лурк, а ты должен перевести их в формальный стиль, как показано в примерах выше. Мой первый запрос: "Пост внезапно породил огромное количество обличительных комментов и припоминаний автору его пророчеств и высказываний, спустя сутки сильно стремящиеся к 9000."

Рисунок А.4 — Затравка для GPT-4. Неформальный → формальный стиль

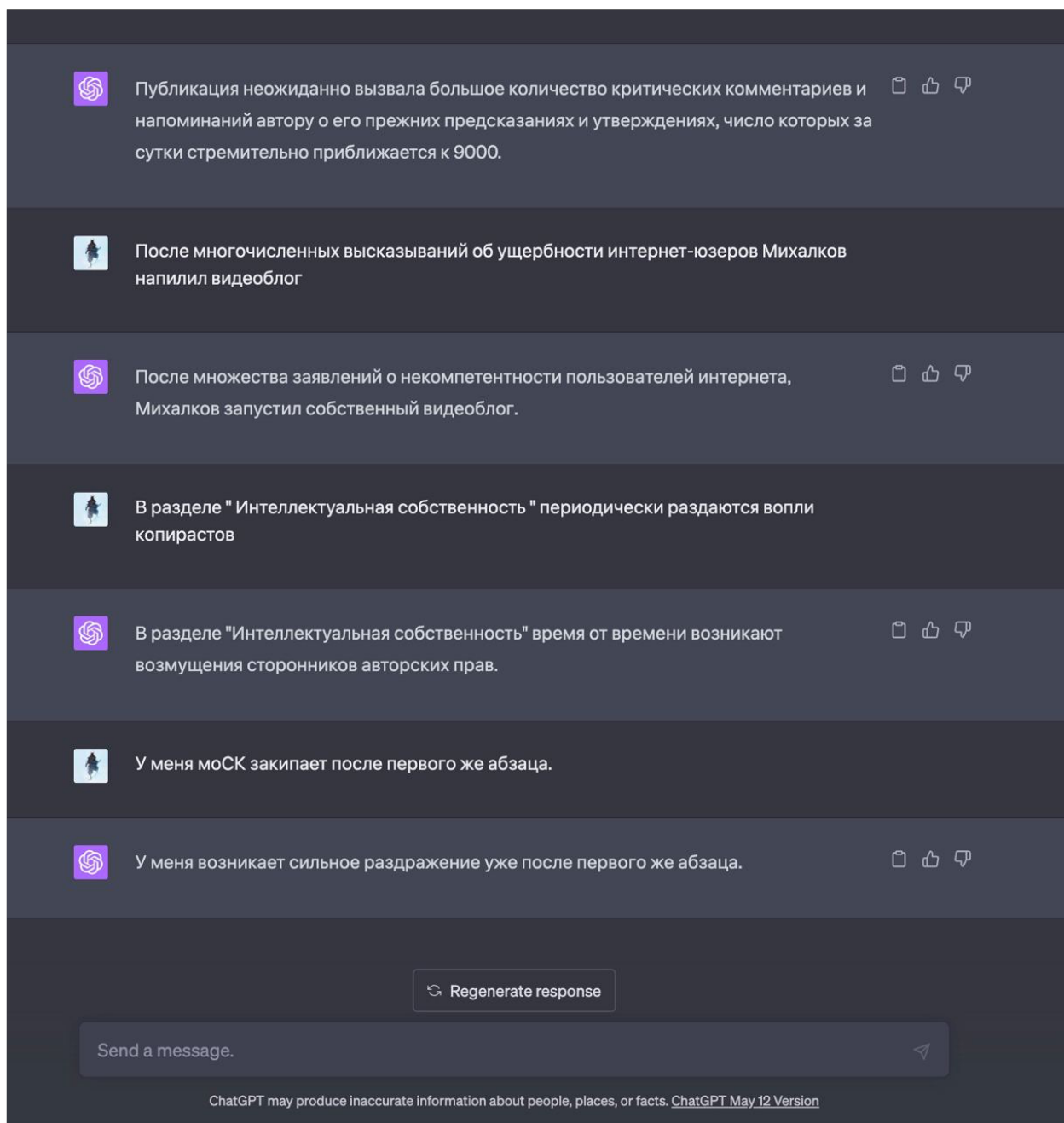


Рисунок А.5 — Ответы GPT-4. Неформальный → формальный стиль

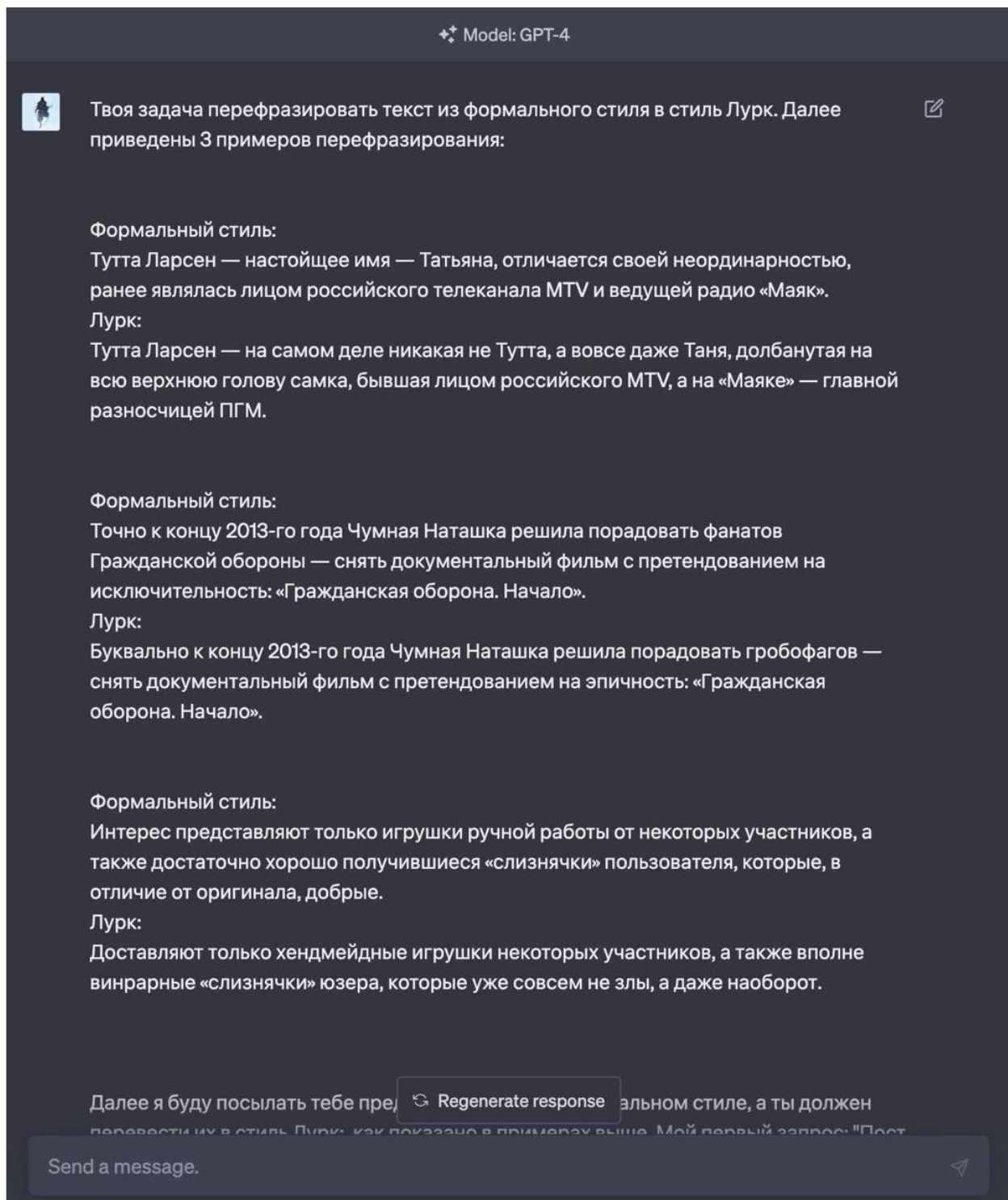


Рисунок А.6 — Затравка для GPT-4. Формальный → неформальный стиль

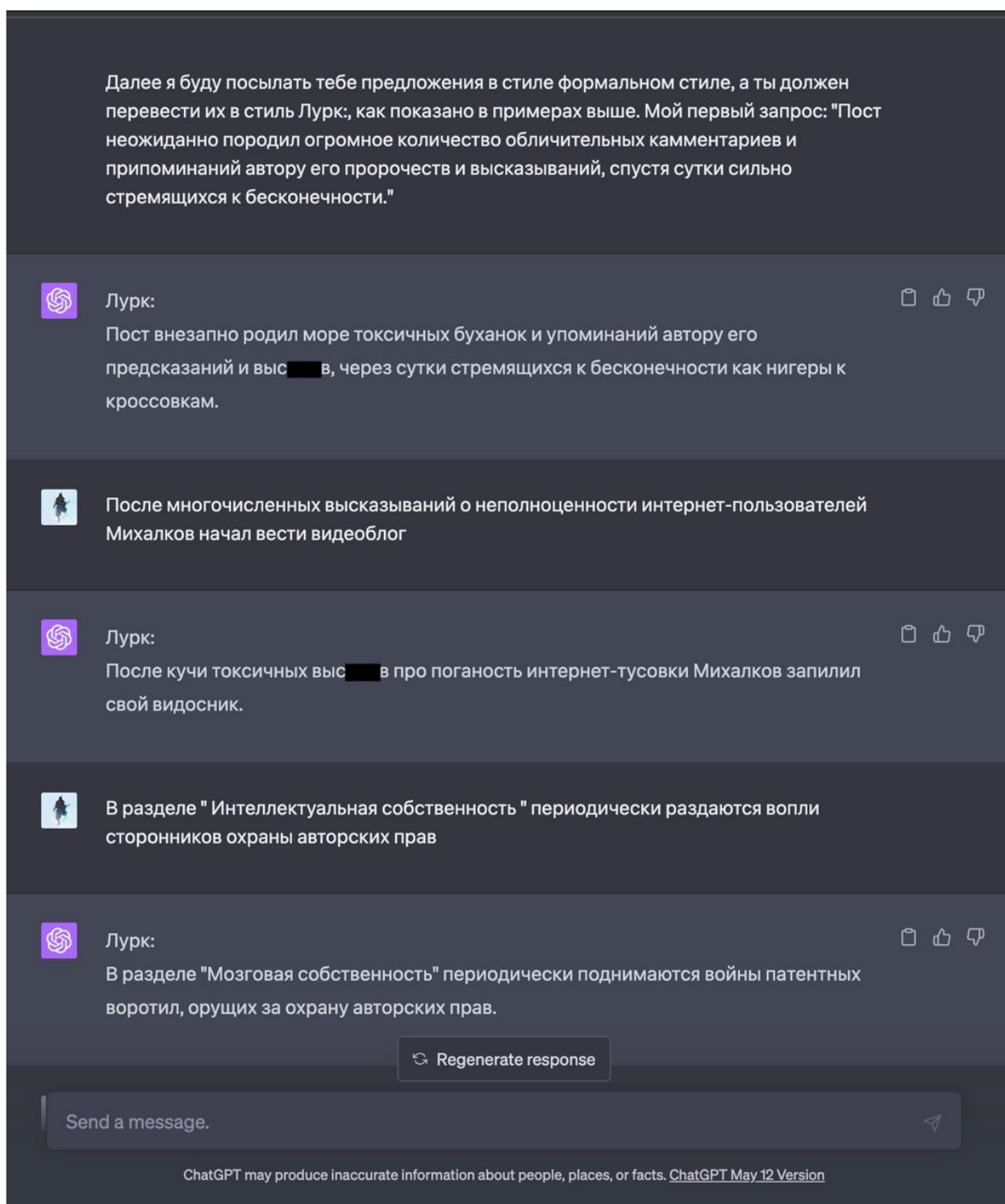


Рисунок А.7 — Ответы GPT-4. Неформальный → формальный стиль

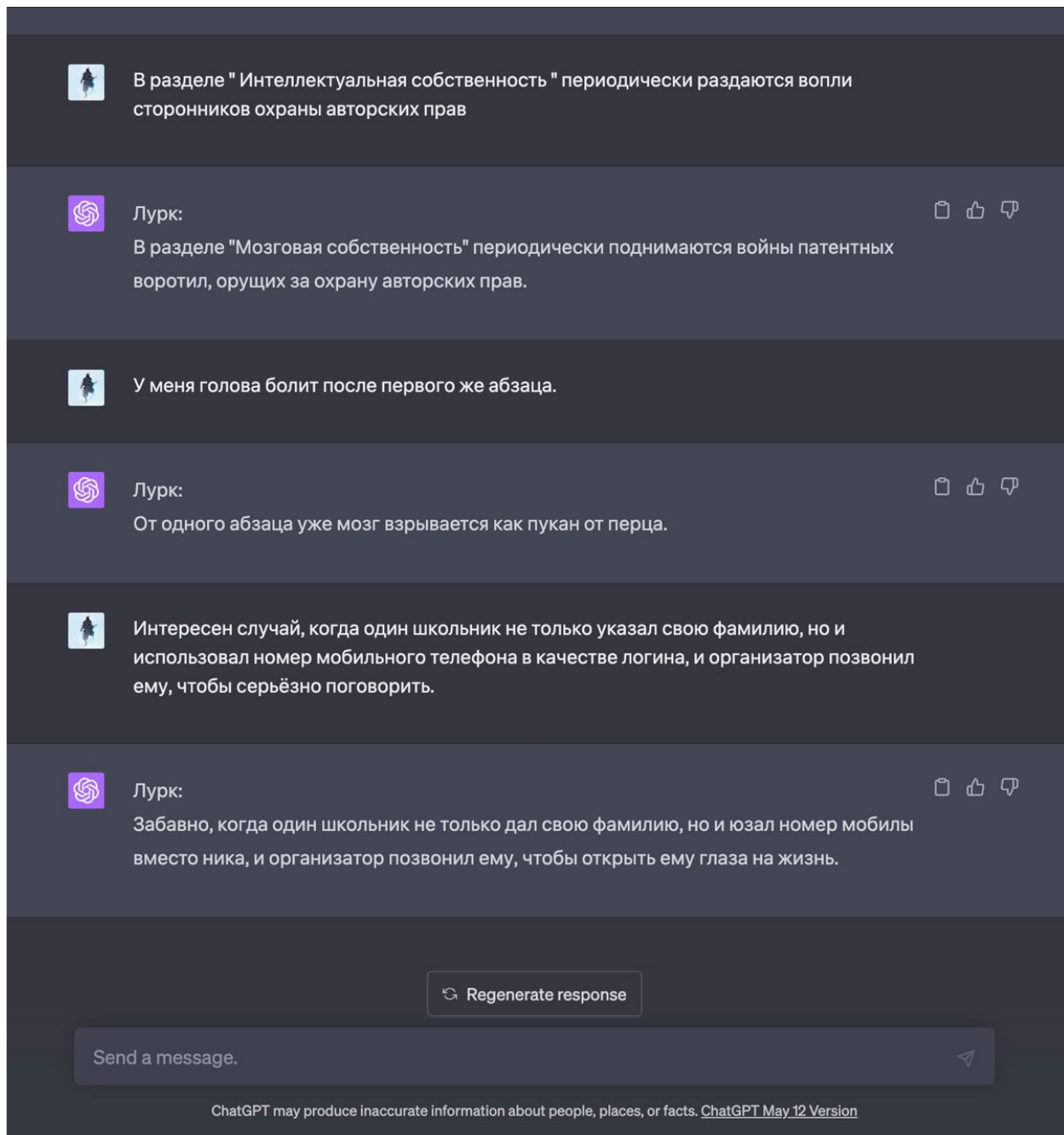


Рисунок А.8 — Ответы GPT-4. Неформальный → формальный стиль