«Московский физико-технический институт (национальный исследовательский университет)»
Физтех-школа прикладной математики и информатики
Кафедра «Интеллектуальные системы»

Пилькевич Антон Александрович

Оптимизация критерия, заданного нейросетевой моделью, в задаче детоксификации текста

03.03.01 – Прикладные математика и физика

Выпускная квалификационная работа бакалавра

Научный руководитель:
д.ф.-м.н. Стрижов Вадим Викторович
Консультант:
Попов Артём Сергеевич

Москва
2022

Содержание

Аннотация

Часто нейросетевые модели используются в качестве метрик качества для задач обработки естественного языка. Возникает желание использовать их в качестве функции потерь, чтобы явно оптимизировать заданную метрику. Но в силу их неизменяемости появляется проблема с дифференцируемостью такой функции потерь, так как различные токенизаторы нарушают её. В первую очередь это связано с тем, что отсутствует инъективное отображение между токенами двух токенизаторов: модели, решающей задачу, и оценочной модели. Данная работа предлагает способ решения отсутствия дифференцируемости такой функции потерь на примере задачи детоксификации текстов. Предлагается обучить новый входной эмбеддинг-слой оценочной модели, который будет принимать распределения вероятностей токенов, выданных моделью детоксификатором. Это позволит использовать её в качестве функции потерь. В работе описывается алгоритм одновременного обучения детоксификатора и адаптера. Также приводятся эксперименты доказывающие эффективность предложенного метода.

Ключевые слова: *deep learning, natural language processing, text style transfer.*

1 Введение

На сегодняшний день огромное число задач машинного обучения решено при помощи нейросетевых моделей. Более того state-of-the-art результаты показывают именно нейросетевые модели и подходы глубокого обучения. Не исключением являются задачи natural language processing (NLP). Например, задачи sentiment analysis, question answering [1], language modelling [2] и другие решены при помощи нейросетевых моделей. В большинстве случаев такие модели имеют десятки или сотни миллиардов параметров, из-за чего их либо очень дорого или просто не возможно обучать с нуля самостоятельно [1, 2]. Также становится всё больше и больше задач, которые в качестве метрик качества используют нейросетевые модели из других задач. Это приводит к желанию использовать эти нейросетевые модели не только в качестве метрик качества, но также в качестве функций потерь, так как это позволит напрямую оптимизировать целевую метрику. Но в задачах NLP возникает проблема связанная с тем, что в большинстве случаев у обучаемой модели и оценочной модели будут различаться токенизаторы. Это приводит к тому, что теряется дифференцируемость такого функционала, так как сперва необходимо получить текст от обучаемой модели при помощи $\arg\max$, далее разбить полученный текст на токены при помощи токенизатора оценочной модели. Обе операции нарушают дифференцируемость, а основная проблема заключается в том, что токены одного токенизатора не могут инъективно отобразиться в токены другого токенизатора.

Схожая проблема, связанная с потерей дифференцируемости, возникает в задачах генерации текстов при помощи порождающих моделей [3, 4]. Дабы решить эту проблему одни подходы используют идеи из Reinforcement Learning [5], основанные на аппроксимации градиентов и методе Монте-Карло. Другие подходы используют идею gumbel-softmax для апроксимации one-hot векторов полученных предложений [6]. Но основная проблема дифференцируемости в этих работах связана с операцией $\arg\max$, в то время как в данной работе всё сводится к различным токенизаторам модели-детоксификатора и оценочной модели.

В данной работе рассматривается задача детоксификации тектов. Требуется для токсичного предложения сгенерировать нейтральную его версию. Под токсичностью в большинстве случаев подразумевается наличие обсценной лексики. Задачу можно рассматривать, как задачу стилизации текстов. В качестве одной из основных метрик качества используется нейросетевая модель — Style Transfer Accuracy (STA), оценивающая степень токсичности предложения. Использование нейросетевой модели приводит к описанным ранее проблемам. Предлагается обучить *адаптер*, который заменит эмбеддинг слой оценочной модели. Адаптер — это линейный слой, который на вход принимает распределения вероятностей токенов, выданных моделью-детоксификатором, и возвращает аппроксимацию входных эмбеддингов оценочной модели. Данная работа описывает алгоритм обучения адаптера и модели-детоксификатора. Благодаря предложенному методу удалось добиться прироста метрики STA на 7% и увеличить целевую метрики на 4%, представляющую собой произведение STA и chrF1 [7]. Также в работе описывается процесс нахождения оптимального алгоритма обучения адаптера и детоксификатора, демонстрируются полученные результаты при использовании различных подходов.

2 Постановка задачи

Решается задача *детоксификации* текстов. Требуется по токсичному предложению построить его нейтральный вариант, который сохраняет семантический смысл. Под токсичностью в большинстве случаев подразумевается наличие обсценной лексики.

2.1 Детоксификация, как машинный перевод

Задачу детоксификации можно рассматривать, как задачу переноса текстового стиля. Но в данной работе имеется параллельный корпус текстов, то сперва удобнее рассматривать задачу, как задачу машинного перевода (seq-to-seq).

Данные представляют собой параллельный корпус текстов (s_t, s_d), где s_t — токсичное предложение из датасета, s_d — его нейтральная версия. Обозначим два пространства с различными стилями: T — пространство токсичных предложений, D — пространство нейтральных предложений. Тогда требуется построить модель из T в D.

Для работы нейросетевых моделей с текстом требуется разбить его на токены. В данной работе для разбиения текста на токены используется BPE-токенизатор [8].

Пускай имеется *словарь* V_f, содержащий все возможные токены для модели-детоксификатора. Тогда τ_f — *токенизатор*, такой что

$$\tau_f : T \cup D \to (V_f)^n,$$

где n — фиксированная длина последовательности токенов. Причём в данной постановке задачи токенизатор τ_f считается неизменным, и берётся уже обученным с фиксированным словарём V_f. Пусть также имеется два предложения $s_t \in T$ и $s_d \in D$. Тогда $t = \tau_f(s_t), d = \tau_f(s_d)$ — их разбиения на токены. Введём *модель-детокисфикатор* f_θ, такую что

$$f_\theta : (V_f)^n \to [0,1]^{n \times |V_f|},$$
$$f_\theta(*|d_{<i}, t) \in [0,1]^{|V_f|},$$

где $f_\theta(*|d_{<i}, t)$ — распределение вероятностей i-го токена при условии токсичного предложения и сгенерированных $i-1$ токене.

В простейшей постановке задачи машинного перевода имеется функция потерь задаётся кросс-энтропией:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{n} \log f_\theta(d_i|d_{<i}, t).$$

Тогда решаемая задача записывается как:

$$\mathcal{L}_{\text{CE}} \longrightarrow \min_\theta .$$

2.2 Детоксификация, как текстовая стилизация

Отметим, что в задаче машинного перевода кросс-энтропийная функция потерь никак не учитывает нейтральность полученных предложений, то есть не учитывается задача стилизации. Это распространённая проблема в задачах переноса текстового стиля при наличии параллельного корпуса текстов [9]. Один из возможных способ учитывать нейтральность полученного предложения — это подсчёт потенциально токсичных и оскорбительных слов. При таком подходе возникает много трудностей. Требуется учитывать различную морфологию, семантику, контекст, а также иметь в наличии список потенциально токсичных и оскорбительных слов. Причём такой подход не будет являться дифференцируемым. Поэтому, учитывая результаты нейросетевых моделей в задачах классификации, наиболее эффективным подходом оценки степени токсичности являются нейросети.

Обозначим нейросетевую модель, отвечающую за оценку токсичности предложения, как g_{toxic}. Далее будем называть её *оценочной моделью*. Как и модели-детоксификатора, оценочная модель имеет фиксированный словарь V_g и фиксированный обученый BPE-токенизатор $\tau_g : T \cup D \to (V_g)^n$. Тогда можно записать:

$$g_{\text{toxic}} : (V_g)^n \to [0, 1].$$

Будем рассматривать выход модель, как вероятность токсичности предложения. Введём детоксифицированное предложение s_{detox}, полученное моделью-детоксификатором для входного токсичного предложения s_t,

$$s_{\text{detox}} = \{\arg\max_{d_i} f_\theta(d_i | d_{<i}, t)\}_{i=1}^{n}.$$

Тогда решение следующей задачи будет явно учитывать задачу переноса стиля:

$$\mathcal{L}_{\text{TP}}(s_{\text{detox}}) = g_{\text{toxic}}\big(\tau_g(s_{\text{detox}})\big) \longrightarrow \min_\theta.$$

2.3 Проблема

При использовании оценочной модели g_{toxic} в качестве функции потерь возникает проблема отсутствия дифференцируемости по параметрам детоксификатора θ. Дифференцируемость нарушается при построении s_{detox} из-за операции $\arg\max$, а также по причине различных токенизаторов у детоксификатора и оценочной модели. Вторая проблема в первую очередь возникает из-за различных словарей V_f, V_g. Нет гарантий, что существует инъективное отображение между токенами τ_f и τ_g. Проще всего увидеть проблему на примере:

$$\textit{ты слишком токсичный} \longrightarrow \begin{cases} [\textit{ты},\ \textit{слишком},\ \textit{токс},\ \textit{ич},\ \textit{ный}], & \text{от } \tau_f, \\ [\textit{ты},\ \#UNK,\ \textit{токсичн},\ \textit{ый}], & \text{от } \tau_g. \end{cases}$$

В последующей главе будет предложен метод позволяющий обойти описанные выше проблемы.

3 Обзор литературы

3.1 Задача стилизации

Задачу текстовой стилизации для параллельного корпуса текстов можно рассматривать, как задачу машинного перевода. Поэтому можно использовать методы и подходы для задачи seq-to-seq. Данная работа в качестве базового решение задачи детоксификации основывается на таком подходе, дообучая трансформер T5 [1]. Но при таком подходе никак не учитывается корректность пар в датасете, точное соответствие заданному стилю и насколько расплывчаты "границы"заданного стиля. Поэтому важной составляющей задачи стилизации является факт, что семантический смысл любого текста может быть выражен в любом стиле [9].

3.2 Автоматическая оценка качества стилизации

Одной из составляющих качества задачи стилизации и машинного перевода является оценка сохранения смысла. Для случая параллельных корпусов текстов хорошо себя показывает chrF [7]. Он зачастую демонстрирует высокую корреляцию с человеческой оценкой качества сохранения смысла [10, 11]. Но данная метрика никак не учитывает стиль получаемых предложений. За соответствие стилю хорошо себя показывают нейросетевые подходы [10]. Но при этом для случая детоксификации русско-язычных текстов такой подход слабо коррелировал с челоческими оценками [11].

3.3 Текстовые порождающие модели

Схожая проблема "потери"дифференцируемости встречается у текстовых порождающих моделях. Проблема заключается в том, что дексриминатор должен работать с последовательностью токенов на входе. Но чтобы получить последовательность токенов необходимо применить операцию $\arg\max$ для выхода генератора, что приводит к отсут-

ствию дифференцируемости. Дабы решить эту проблему одни подходы предлагают использовать reinforcement learning [5], gumbel-softmax [6] и другие "ухищрения" [3, 4]. При этом отличительная особенность проблемы дифференцируемости в данной работы заключается в том, что параметры, словарь токенов V_g и токенизатор τ_g оценочной модели g_{toxic} фиксированы, а также зафиксирован словарь V_f и токенизатор τ_g детоксификатора f_θ. Наиболее простым подходом для реализации в данной работы является RL-подход, но зачастую он показывает плохие результаты для порождающих моделей и сложен в обучении. Поэтому был предложен подход с использованием адаптера, который будет описан в следующей главе.

4 Адаптер

4.1 Адаптер — аппроксимация векторного представления

Как отмечалось в главе 2.3, главная проблема при использовании оценочной модели g_{toxic} в качестве функции потерь является отсутствие дифференцируемости по параметрам детоксификатора θ. Предлагается избавиться от явной генерации нейтрального предложения s_{detox} детоксификатором для оценки степень токсичности получившегося предложения.

Введём произвольное предложение $s \in T \cup D$ и операцию $\textbf{one-hot}_g$ такую, что

$$\begin{gathered}\textbf{one-hot}_g : V_g \to \{0,1\}^{|V_g|}, \\ \left(\textbf{one-hot}_g(i)\right)_{j=i} = 1, \\ \left(\textbf{one-hot}_g(i)\right)_{j\neq i} = 0.\end{gathered}$$

Также введём матрицу входных эмбеддингов оценочной модели E_g. *Эмбеддингом токена* будем называть его векторное представление, причём $E_g \in \mathbb{R}^{|V_g|\times e_g}$, где e_g — размер входных эмбеддингов оценочной модели.

Оценочная модель g_{toxic} принимает на вход токены $\tau_g(s)$ сопоставляя им эмбеддинги из матрицы эмбеддингов E_g. Эту операцию можно записать следующим образом:

$$\begin{gathered}O_g E_g \in \mathbb{R}^{n\times e_g}, \\ O_g = \left(\textbf{one-hot}_g\left(\tau_g(s)_i\right)\right)_{i=1}^{n} \in \{0,1\}^{n\times|V_g|}.\end{gathered}$$

Выход детоксификатора $f_\theta\left(\tau_f(s)\right) \in [0,1]^{n\times|V_f|}$, как было описано в главе 2.1, есть распределение вероятностей токенов из V_f. Введём *адаптер* $A \in \mathbb{R}^{|V_f|\times e_g}$, апроксимирущий входные эмбеддинги оценочной

модели следующим образом:

$$f_\theta\big(\tau_f(s)\big)A \approx O_g E_g.$$

При фиксированных параметрах оценочной модели это эквиваленто записи:

$$g^*_{\text{toxic}}\big(f_\theta\big(\tau_f(s)\big)\big) \approx g_{\text{toxic}}\big(\tau_g(s)\big),$$
$$g^*_{\text{toxic}} : [0,1]^{n\times|V_f|} \to [0,1],$$

где g^*_{toxic} — оценочная модель g_{toxic}, в которой заменили входную матрицу эмбеддингов E_g на адаптер A. Иллюстрация схемы работы изображена на рис. 1.

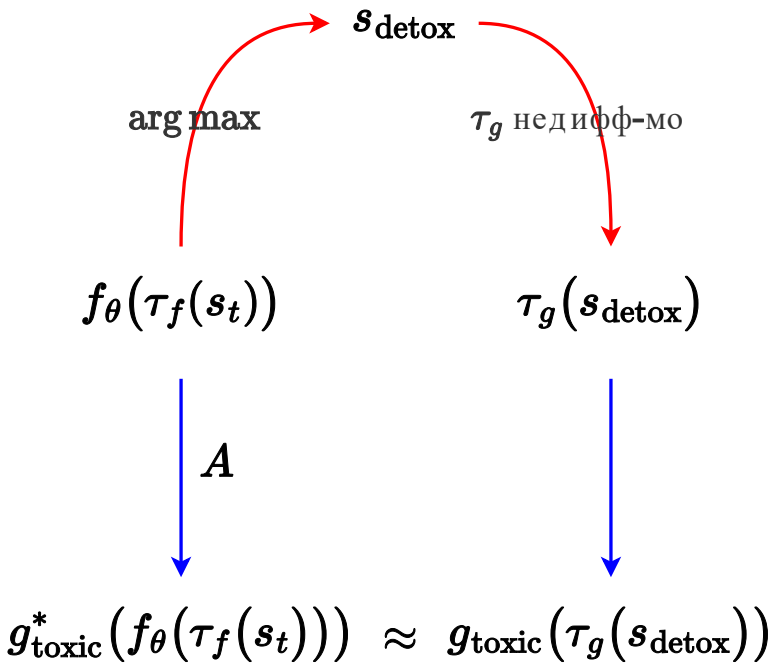

Рис. 1: Схема работы адаптера и решение проблемы дифференцируемости. Синими стрелками изображены операции не нарушающие дифференцируемость, красными - нарушающие.

В таком подходе можно интерпретировать выходное распределение детоксификатора $f_\theta\big(\tau_f(s)\big)$, как "зашумлённые"one-hot вектора. Так как матрица эмбеддингов до этого принимала вектора со значениями

в $\{0, 1\}$, а теперь принимает со вектора значениемями на непрерывном отрезке $[0, 1]$.

4.2 Обучение адаптера

Будем рассматривать две модели в этой главе: g_{toxic} — исходная оценочная модель с фиксированными параметрами, g^*_{toxic} — оценочная модель g_{toxic}, в которой заменили входную матрицу эмбеддингов E_g на адаптер A, и принимающая распределние токенов, выданных детоксификатором. В этой главе обучаемые параметры — это только параметры адаптера A. Также будем считать, что имеется обученная на задачу машинного перевода детоксификатор f_θ, как это было описанно в главе 2.1.

Пусть имеется пара предложений s_t, s_d — токсичное и его нейтральный вариант соответственно. Введём две вероятности токсичности:

$$\begin{aligned} P_{\text{toxic}} &= g_{\text{toxic}}\big(\tau_g(s_d)\big), \\ P^*_{\text{toxic}} &= g^*_{\text{toxic}}\big(f_\theta\big(\tau_g(s_t)\big)\big). \end{aligned}$$

Тогда определим функцию потерь для обучения адаптера, как дивергенцию Кульбака-Лейблера:

$$D_{\text{KL}}\left(P_{\text{toxic}} \parallel P^*_{\text{toxic}}\right) = P_{\text{toxic}} \cdot \log\left(\frac{P_{\text{toxic}}}{P^*_{\text{toxic}}}\right) + (1 - P_{\text{toxic}}) \cdot \log\left(\frac{1 - P_{\text{toxic}}}{1 - P^*_{\text{toxic}}}\right).$$

И при обучении адаптера требуется решить следующую задачу:

$$D_{\text{KL}}\left(P_{\text{toxic}} \parallel P^*_{\text{toxic}}\right) \longrightarrow \min_A.$$

Схема обучения адаптера изображена на рис. 2.

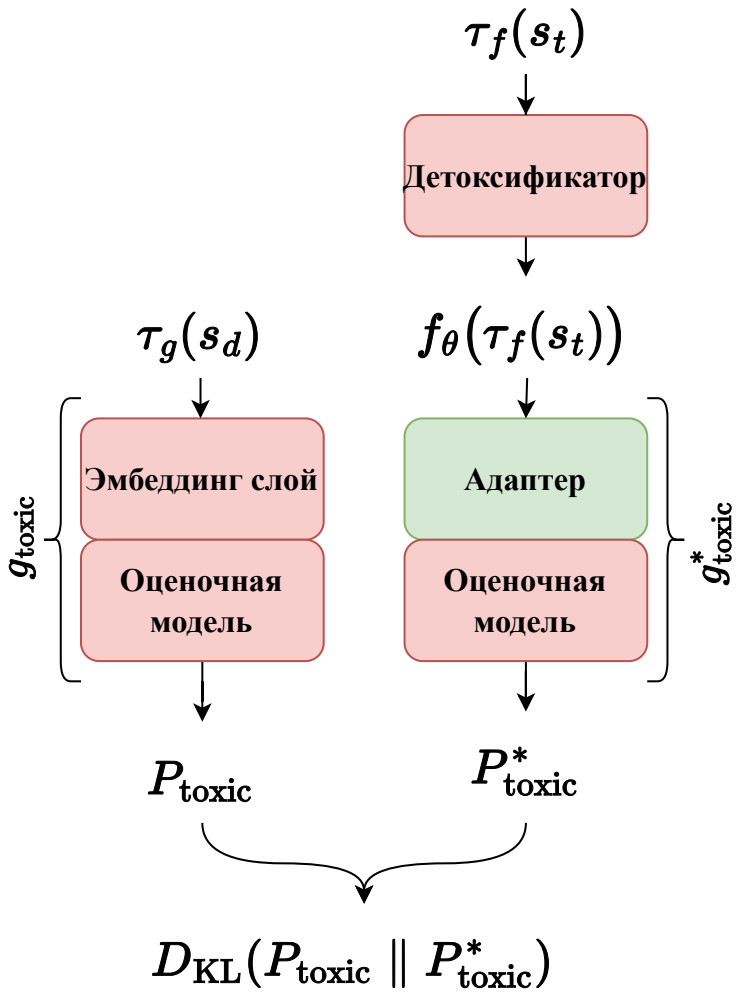

Рис. 2: Схема обучения адаптера.

4.3 Использование адаптера в дообучении детоксификатора

В этой главе будем считать, что имеется обученная модель детоксификатора f_θ на задачу машинного перевода и адаптер A для модели g^*_{toxic}. Отметим, что адаптер был обучен для фиксированных параметров детоксифакатора. Поэтому при дообучении детоксификатора будет инвалидироваться результат обучения адаптера. Дабы избежать этого, требуется модифицировать алгоритм дообучения детоксификатора. Основная идея обучения детоксификатора основана на обучении порождающих моделей [12].

Введём следующие обозначения:

$$\text{CE} = \mathcal{L}_{\text{CE}},$$
$$\text{TP} = P^*_{\text{toxic}}.$$

Введём *агрегирующую функцию потерь* $\mathcal{F} : \mathbb{R}^2 \to \mathbb{R}$, которая будет принимать значения кросс-энтропии $\mathcal{L}_{\text{CE}}$ и вероятности токсичности P^*_{toxic}. Тогда предложенный алгоритм можно описать следующим образом:

1. N батчей обучается детоксиификатор f_θ, минимизирую функцию потерь $\mathcal{F}(\text{CE}, \text{TP})$, при фиксированных параметрах адаптера A.

2. M батчей обучается адаптер A, минимизирую D_{KL}, как в главе 4.2, при фиксированных параметрах детоксификатора f_θ.

5 Вычислительный эксперимент

5.1 Данные

Данные взяты из соревнования *"Russian Text Detoxification Based on Parallel Corpora"* [13]. Предложения были собраны с таких платформ, как "Одноклассники"[1], "Pikabu"[2], "Twitter"[3]. Как было описано раннее, датасет представляет себе параллельный корпус русскоязычных предложений, состоящий из токсичного варианта и его перефразированного нейтрального варианта. Нейтральная версия токсичного предложения была получена при помощи толокерской разметки[4]. У каждого токсичного предложения имеется от одного до трёх нейтральных вариантов.

Размер обучающего датасета составляет $N_{\text{train}} = 11136$ пар (s_t, s_d). Из которых 10% использовались для валидации во время обучения. Размер тестового датасета составляет $N_{\text{test}} = 800$ предложений. Примеры:

Токсичное предложение	Нейтральный вариант
«Её муженька козла на кол надо посадить.»	*«Её мужа нужно наказать.»*
«Это твари а не люди»	*«Это плохие люди.»*
«Да такой же урод и выдал.»	*«Этот же человек и посоветовал.»*

5.2 Метрики

В качестве метрик для оценки качества детоксификации использовались chrF [7], Style transfer accuracy (STA) и их произведение.

ChrF — F-score на основе символьных n-грамм. Отвечает за оценку качества «перевода» по сравнению с истинным предложения. Введём chrP — доля символьных n-грамм из предлагаемого предложения, которые имеются в оригинальном. И chrR — доля символьных n-грамм

[1] https://www.kaggle.com/datasets/alexandersemiletov/toxic-russian-comments

[2] https://www.kaggle.com/datasets/blackmoon/russian-language-toxic-comments

[3] http://study.mokoron.com/

[4] https://toloka.yandex.ru/

из оригинального предложения, которые представлены в предлагаемом. Тогда финальная формула:

$$\text{chrF}_\beta = (1 + \beta^2)\frac{\text{chrP} \cdot \text{chrR}}{\beta^2\text{chrP} + \text{chrR}},$$

где chrP, chrR считаются как среднее по всем предложениям.

Style transfer accuracy (STA) — вероятность токсичности предложения. Метрика задаётся оценочной моделью g_{toxic}. В качестве модели g_{toxic} используется предобученный Conversational RuBERT, дообученный на задачу определения токсичности[5].

Тем самым chrF1 отвечает за сохранения смысла по сравнению с истинным нейтральным предложением, а STA отвечает за перенос стиля. В качестве целевой метрики используется произведение chrF1 и STA, дабы учесть влияние каждой из задач.

5.3 Baseline T5

Как было описано в главе 2.1, при наличии параллельного корпуса текстов задачу текстовой стилизации можно решать, как задачу машинного перевода. Поэтому в качестве исходной точки используется именно такой подход, чтобы была возможность оценивать эффект предложенных подходов.

В качестве детоксификатора f_θ используется трансформер T5 [1], то есть архитектура кодировщик-декодировщик. Так как размер датасета в данной работе мал, то наиболее выгодным решением является использование предобученную архитектуры и дообучение её на задачу детоксификации. В качестве предобученной модели берётся ruT5-base[6], обученная на задаче seq-to-seq при испольховании датасет размером 300Gb.

[5] `https://huggingface.co/SkolkovoInstitute/russian_toxicity_classifier`

[6] `https://huggingface.co/sberbank-ai/ruT5-base`

5.4 Одновременное обучение адаптера и детоксификатора

В этой и следующей главе в качестве исходной модели детоксификатора f_θ используется модель полученная в предыдущей главе 5.3, после дообучения на задачу детоксификации, как на задачу машинного перевода.

Изначальная идея метода заключалась в одновременном обучении адаптера A и модели детоксификатора f_θ. Причём адаптер на вход принимал выходные эмбеддинги модели детоксификатора. Решаемую в это случае задачу можно записать через агрегирующую функцию потерь $\mathcal{F}$ следующим образом:

$$\mathcal{F}(\text{CE},\ \text{TP}) \longrightarrow \min_{\theta, A}.$$

Но такой подход естественным образом приводит к переобучению адаптера. Адаптеру выгодно возвращать "шум"для оценочной модели g_{toxic^*}, чтобы уменьшить значения стилизационной функции потерь $\mathcal{L}_{\text{TP}}$. Как видно из таблицы 1, данный подход приводил к переобучению. Целевая метрика STA*chrF1 ухудшалась на 8%, что сильнее всего связано с уменьшением STA-метрики на 10%.

Следующая идея заключалась в использование softmax от выходных логитов модели детоксификатора f_θ в качестве входов адаптера при обучении на одинаковую функцию потерь $\mathcal{F}$. В этом случае выход детоксификатора можно интерпретировать, как one-hot вектора детоксификатора, как это было описано в главе 4.1. Но также, как и в предыдущем эксперименте, такой подход обучения детоксификатора приводит к переобучению модели, что видно из результатов в таблице 1. При этом результат получается лучше, чем при использовании эмбеддингов, что связано с меньшей "свободой"для адаптера и возможностей для переобучения.

В обоих подходов в качестве агрегирующей функции потерь $\mathcal{F}$ использовалось произведение: $\mathcal{F}(\text{CE},\ \text{TP}) = \text{CE} \cdot \text{TP}$. Данный выбор был обусловлен тем, что целевой метрикой является STA*ChrF1.

Подход	$\mathcal{F}$	STA	chrF1	STA*chrF1
seq-to-seq	CE	0.742	0.577	0.428
Adapter on embs	CE $\cdot$ TP	0.639	0.544	0.348
Adapter on logits	CE $\cdot$ TP	0.708	0.569	0.403

Таблица 1: Результаты подхода, когда детоксификатор и адаптер оптимизируют общую функцию потерь $\mathcal{F}$.

5.5 Удачные эксперименты

Следующий подход заключался в раздельном обучении модели детоксификатора и адаптера при использовании различных функций потерь. Сперва обучался адаптер A также, как это было описано в главе 4.2. Но далее, в отличии от финальной версии предложенного метода, на одном и том же батче сперва обучался детоксификатор при фиксированном адаптаре на $\mathcal{F}$, а потом дообучался адаптер при фиксированном детоксификаторе на D_{KL}. Далее такой подход будем называть как *Same batch*.

В качестве функции потерь $\mathcal{F}(\text{CE}, \text{TP})$ использовались два варианта:

$$\mathcal{F}(\text{CE}, \text{TP}) = \text{CE} \cdot \text{TP},$$
$$\mathcal{F}(\text{CE}, \text{TP}) = w_1 \cdot \text{CE} + w_2 \cdot \text{TP}.$$

Наилучшим образом себя показали коэффициенты для взвешенной функции потерь: $w_1 = 1, w_2 = 10$. Это объясняется тем, что модель детоксификатор до этого уже была обучена на кросс-энтропию и важно обращать внимание на функцию потерь, отвечающую за стилизацию, и взвешенная функция потерь позволяет сделать это явно. В данной постановке обучения проводились три запуска каждого из экспериментов с различными seed. Как видно из таблицы 2, есть значимое улучшение STA на 3%, но при этом происходит уменьшение chrF1. В результате чего целевая метрика в виде их произведение почти не изменяется. При этом использование различных функций потерь $\mathcal{F}$ даёт практически одинаковый результат.

Подход	$\mathcal{F}$	STA	chrF1	STA*chrF1
seq-to-seq	CE	0.744 ± 0.010	0.575 ± 0.002	0.430 ± 0.042
Same batch	$\text{CE} \cdot \text{TP}$	0.776 ± 0.015	0.569 ± 0.004	0.442 ± 0.006
Same batch	$\text{CE} + 10 \cdot \text{TP}$	0.774 ± 0.011	0.561 ± 0.012	0.435 ± 0.013

Таблица 2: Эксперименты по проверке статистической значимости подхода с обучением детоксификатора и адаптера на одном батче при использовании разных функций потерь, Same batch.

После этого использовалось финальный подход обучения, предложенная в главе 4.3. Будет называть его *GAN style*. Также как и в предыдущем эксперименте использовались два различных варианта функции потерь $\mathcal{F}$:

$$\mathcal{F}(\text{CE}, \text{TP}) = \text{CE} \cdot \text{TP},$$
$$\mathcal{F}(\text{CE}, \text{TP}) = w_1 \cdot \text{CE} + w_2 \cdot \text{TP}.$$

При таком подходе обучения наилучшим образом себя показал вариант с взвешенной функцией потерь и явным предпочтением для стилизационной функции потерь: $w_1 = 1, w_2 = 10$. Результаты отображены в таблице 3. Удалось достичь улучшения STA на 7.4% и STA*chrF1 на 3.5%.

Также проводился эксперимент с дообучение детоксификатора только на стилизационную функцию потерь TP. При таком подходе происходит явное переобучение, что видно из метрик: STA ≈ 1.0 и ChrF1 в 5 раз меньше. Если посмотреть на сгенерированные предложения, то окажется, что детоксификатор выдаёт одно и тоже предложение для любого входа.

Подход	$\mathcal{F}$	STA	chrF1	STA*chrF1
seq-to-seq	CE	0.739	0.578	0.427
GAN style	TP	0.998	0.119	0.119
GAN style	$\text{CE} \cdot \text{TP}$	0.754	0.574	0.439
GAN style	$\text{CE} + 10 \cdot \text{TP}$	**0.813**	0.569	**0.462**

Таблица 3: Что то

6 Заключение

Данная работа предлагает метод использования нейросетевых моделей в качестве функции потерь в задачах обработки естественного языка. При этом возникает проблема деффиринцируемости такой функции потерь из-за различных токенизаторов у модели детоксификатора и оценочной модели. Предложенный метод доказывает свою работоспособность и эффективность. Также опубликован код для воспроизведение финального подхода[7]

В последующих работах необходимо обобщить предложенный метод на другие оценочные модели. Также стоит проверить работоспособность метода в других задачах. Отдельный интерес представляет сравнение с методами из текстовых порождающих моделей в текущей постановки задаче, так и сравнение предложенного метода в задачи генерации текста против текущих подходов порождающих моделей.

[7] https://github.com/Intelligent-Systems-Phystech/Pilkevich-BS-Thesis

Список литературы

[1] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 33:1877–1901, 2020.

[3] Danilo Croce, Giuseppe Castellucci, and Roberto Basili. Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples. pages 2114–2119, jul 2020.

[4] David Donahue and Anna Rumshisky. Adversarial text generation without reinforcement learning. *CoRR*, abs/1810.06640, 2018.

[5] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[6] Matt J Kusner and José Miguel Hernández-Lobato. Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, 2016.

[7] Maja Popović. chrf: character n-gram f-score for automatic mt evaluation. pages 392–395, sep 2015.

[8] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[9] Alexey Tikhonov and Ivan P. Yamshchikov. What is wrong with style transfer for texts? *ArXiv*, abs/1808.04365, 2018.

[10] Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1321–1336, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[11] Varvara Logacheva, Daryna Dementieva, Irina Krotova, Alena Fenogenova, Irina Nikishina, Tatiana Shavrina, and Alexander Panchenko. A study on manual and automatic evaluation for text style transfer: The case of detoxification. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 90–101, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[13] `https://russe.nlpub.org/2022/tox/`. Russian Text Detoxification Based on Parallel Corpora.