Генеративно-состязательный подход в обработке естественного языка

Е. Н. Каруна, П. В. Соколов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
zhenya.karuna@mail.ru, pvsokolov@etu.ru

Д. А. Гаврилик
Санкт-Петербургский политехнический университет Петра Великого
daria.gavrilic@gmail.com

***Аннотация*. Применение генеративно-состязательного алгоритма обучения нейронных сетей позволило значительно продвинуться в решении задачи генерации изображений и аудиоданных. Тем не менее, остаются важные проблемы при решении задач генерации дискретных последовательностей данных. Решение подобных проблем позволит использовать генеративно-состязательное обучение для генерации текстовых данных. В данной работе отражён краткий обзор современных исследований и достижений в генерации текстовых данных с помощью генеративно-состязательного обучения, перечислен набор задач, которые можно решить с помощью данного подхода, выполняется описание возможных проблем и существующих способов решения имеющихся проблем, а также описываются некоторые предложения по улучшению моделей. Описывается структура и алгоритм предложенной системы, приводятся результаты исследований.**

Ключевые слова: генеративно-состязательное обучение; генерация парафраз; перефразирование; обработка естественного языка

I. Введение

Задача генерации данных была одной из самых сложных задач в машинном обучении. Это привело к тому, что были разработаны качественные модели, способные сопоставить многомерный вход с метками класса, но отсутствовали модели, способные качественно генерировать изображения, аудио, видео или текст. Большой скачок в решении задачи генерации удалось совершить благодаря вышедшей в 2014 году статьи «Generative Adversarial Nets» [1], которая описывала генеративно-состязательные сети.

В данной работе сделан обзор возможностей и способов реализации генеративно-состязательного обучения для обработки текстовых данных, приводятся описание и актуальность задачи разработки модели перефразирования и приводятся результаты первого этапа исследований модели перефразирования на основе генеративно-состязательного обучения.

II. Применение генеративно-состязательных сетей

A. Описание генеративно-состязательного обучения

Генеративно-состязательная сеть (*англ.* Generative Adversarial Nets – GAN) является комбинацией двух нейронных сетей: генератора *G* и дискриминатора *D*. Задачей генератора является создание объекта $\hat{x}$ из множества $\hat{X}$, причём свойства объекта $\hat{x}$ должны быть близки к объекту x из множества X настолько, чтобы дискриминатор *D* не смог их отличить. В это же время дискриминатор D представляет собой бинарный классификатор. Он получает на вход объекты из множеств X, $\hat{X}$. Задачей классификатора является определить, к какому из двух множеств, принадлежит входящий объект. Таким образом, архитектура генеративно-состязательной сети предполагает состязание двух моделей, и основной задачей этой архитектуры является обучение генератора порождать объекты $\hat{x}$, близкие по свойствам к объектам множества X.

Изменение выхода генератора $\hat{x}$ обеспечивается изменением входящего в генератор вектора шума z. Вероятностное распределение генератора p_g над набором данных X должно соответствовать вероятностному распределению p_{data}. Таким образом, генератор можно представить как отображение $G(z,\gamma_g): Z \to \hat{X}$, где G – дифференцируемая функция, представленная параметрами γ_g. В то время как дискриминатор можно представить как $D(x,\gamma_d)$, где γ_d – параметры дискриминатора.

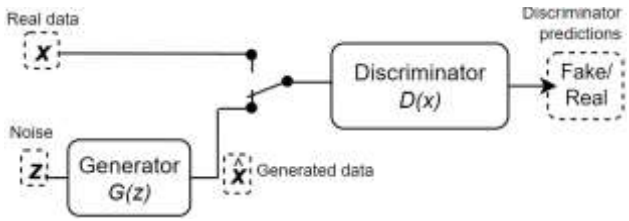

Рис. 1. Оригинальная архитектура генеративно-состязательной сети

Обучение генератора заключается в максимизации вероятности ошибки $D(x,\gamma_d)$ при изменении только параметров γ_g, в то время как обучение дискриминатора состоит в минимизации вероятности ошибки $D(x,\gamma_d)$ при возможности изменения только параметров γ_d. Таким образом, получается минимакс игра, описываемая функцией:

$$\min_G \max_D V(D,G) = \mathrm{E}_{x\sim p_{data}}[\log D(x)] + \\ +\mathrm{E}_{z\sim p_z}[\log(1-D(G(z)))], \quad (1)$$

где V – функция потерь, $\mathbb{E}_x$ – математическое ожидание по всей выборке из реального набора данных, $\mathbb{E}_z$ – математическое ожидание по всем входным данным *z*.

Лучших результатов такой подход достиг в генерации изображений высокого разрешения, задачах переноса стиля изображения. Существующее разнообразие архитектур велико, и направление продолжает активно развиваться [2, 3].

B. *Перспективы использования генерации текстовой информации с помощью состязательного подхода*

Применение генеративно-состязательного обучения для порождения текстовых данных может обладать рядом преимуществ по сравнению с другими способами обучения. Одним из таких, на наш взгляд, является нацеленность генеративно-состязательного подхода порождать большое разнообразие ответов на любой входной запрос. Многообразие форм языка и способов описания одного и того же явления разными способами на интуитивном уровне подсказывает, что при обучении моделей генерировать текст некорректно подавать ей на выход уже готовый текст, который она будет использовать для корректировки своих весов. Дело в том, что подобный подход, так или иначе, заставляет модель выучивать один конкретный набор ответов на входные данные. В то время как набор возможных ответов, как правило, может быть неограниченно большим, и зависит он от языковых особенностей, и особенностей речи говорящего.

Помимо этого, применение генеративно-состязательных сетей для генерации текстовых данных является малоизученной и достаточно перспективной областью. Существует некоторое количество исследований, в которых GAN применяется для решения задач машинного перевода, ведения диалога, продолжения текста, ответов на вопросы и генерации текстов с заданной эмоциональной окраской [4].

Возможной перспективной целью может стать задача построения модели перефразирования с помощью генеративно-состязательного обучения. Модель должна получать на вход исходный текст и на его основе генерировать парафразу. Из-за отсутствия примеров правильных парафраз она должна быть способна порождать новые ответы, которые будут отличаться от обучающего набора данных. Решение подобной задачи позволит улучшить способность языковых моделей к пониманию языковых особенностей.

Задачей модели перефразирования *M* является генерация текста $\hat{x}$ из исходного текста x, такого что тексты x и $\hat{x}$ будут максимально близки друг другу по смыслу, но различны по способу написания. Однако, для построения подобной модели необходимо решить ряд проблем, связанных с обработкой текстовых данных, характерных для генеративно-состязательных сетей, речь о которых пойдёт в следующей главе доклада.

III. Проблемы генерации текстов

Одним из важных свойств, присущих алгоритмам генерации изображений на основе глубокого обучения, является то, что порождаемые данные имеют непрерывное распределение вероятности. Подобным свойством не обладают текстовые данные, так как текст по своей форме – это дискретная последовательность элементов (слов) из ограниченного множества элементов (словаря). Процесс генеративно-состязательного обучения требует передачи градиентов ошибок от дискриминатора к генератору для решения задачи оптимизации параметров генератора. Следовательно, операторы всех слоев нейронной сети должны быть дифференцируемы. В то время как языковые модели содержат оператор выбора категориального значения (токенов) *argmax* из функции *softmax*, который не обладает необходимым свойством.

Для решения проблем с передачей ошибок от дискриминатора к генератору существует несколько возможных решений [4]:

- дифференцирование с помощью распределения *Gumbel-softmax*. Для обхода недифференцируемого оператора выбора *argmax* можно использовать непрерывное распределение *Gumbel-softmax*, которое способно его аппроксимировать [7][8];
- оптимизация параметров генератора на основе обучения с подкреплением. Предлагается получить сигналы вознаграждения от дискриминатора и передать их обратно генератору так, как это бы сделал метод обучения с подкреплением [5][6];
- обучение в непрерывном скрытом пространстве. Помимо использования специальных техник для передачи градиентов через набор дискретных элементов, можно выполнять обучение моделей полностью в скрытом векторном пространстве.

IV. Предложения для новых исследований

Из множества задач области обработки естественного языка задача перефразирования является именно такой, решение которой остаётся неудовлетворительным. Связано это, в первую очередь с тем, что в таких задачах необходимо учитывать сложную семантику на более высоком уровне, нежели для других задач обработки естественного языка. В качестве одного из подходов предлагается применить GAN для решения данной задачи. Генеративно-состязательные сети позволяют порождать большое разнообразие сгенерированных данных и, при этом, они стремятся генерировать данные не путём аппроксимации всех входных примеров, а путём моделирования нового распределения данных, которое зависит от дискриминатора. Такой подход может значительно увеличить разнообразие данных, что является ключевым параметром для задачи перефразирования.

Разработка модели перефразирования сводится к обучению модели, на вход которой поступает исходный текст x_a, а на выходе парафраза этого текста $\widehat{x_a}$. На вход дискриминатора должен поступать и исходный текст x_a, и парафраза этого текста x_b или $\widehat{x_a}$. В зависимости от источника парафразы, выход дискриминатора должен вернуть нужную метку класса. Источником парафразы может выступать соответствующий исходному тексту экземпляр парафразы из обучающего набора данных x_b или парафраза, порождённая генератором $\widehat{x_a}$.

$$D(x_a, x_b) = real; \quad D(x_a, \hat{x}_a) = fake, \qquad (1)$$

где x_a – исходный текст, x_b – парафраза из обучающего набора данных, $\hat{x}_a$ – парафраза, порождённая генератором.

С учетом (2) уравнение (1) получит небольшие изменения:

$$\min_G \max_D V(D,G) = \mathrm{E}_{x\sim p_{data}}[\log D(x_a, x_b)] + \\ + \mathrm{E}_{z\sim p_z}[\log(1 - D(x_a, G(x_a)))]. \qquad (2)$$

Одной из проблем данного подхода является дискретный характер текстовых данных. Каждое слово или буква представлены одним элементом из конечного множества слов или букв. Оператор выбора индекса слова из выходного вектора генератора, является недифференцируемым. Для минимизации своей ошибки генератор, методом обратного распространения ошибок, получает от дискриминатора градиент ошибок, передающийся от функции потерь дискриминатора. Наличие хотя бы одного недифференцируемого звена в сети приводит к прекращению распространения градиента ошибок и, соответственно, к невозможности обучать генератор. Для обхода данной проблемы существует несколько подходов. Одним из таких является выполнение всех операций обучения на уровне векторов скрытого состояния моделей. Таким образом, выходом генератора будет скрытый вектор, который не будет преобразовываться в текст, а пойдёт сразу на вход модели дискриминатора, обходя этап преобразования текста в вектор.

В качестве моделей генератора и дискриминатора была использована модель *T5* [10] архитектуры *transformer*. В модели дискриминатора был убран декодер и добавлена сеть прямого распространения, решающую задачу бинарной классификации. Структурная схема модели перефразирования представлена на рис. 2.

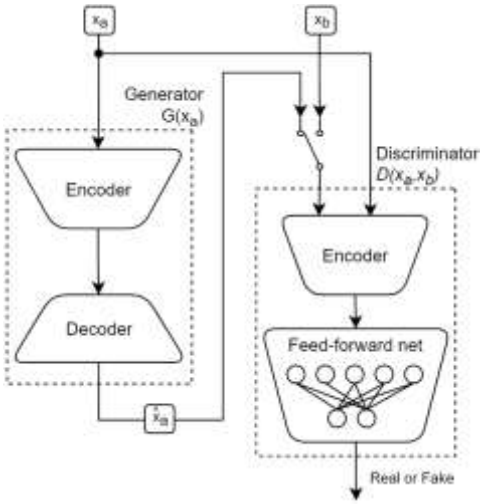

Рис. 2. Предложенная структурная схема модели генеративно-состязательного обучения. Где x_a – исходный текст, x_b – парафраза исходного текста, $\widehat{x_a}$ – парафраза, порожденная генератором

При обучении и тестировании было решено упростить работу модели, и для обучения использовалась не коллекция собранных парафраз, а искусственно созданный набор парафраз. Для генерации парафраз использовался метод генерации на основе правил, замены синонимов и перестановок слов. В качестве исходных текстов использовался набор русскоязычных новостных статей из открытых источников, а на вход модели подавались тексты по одному предложению. Подобный подход к выбору обучающего набора данных является наиболее подходящим для первого этапа разработки модели, так как количество данных для обучения становится неограниченным, а также по причине простоты сгенерированных парафраз. На этапе разработки и отладки целесообразно использовать наиболее простой набор парафраз для ускорения обучения и оптимизации процесса отладки моделей.

V. Результаты

Основными целями в практической части работы является построение генеративно-состязательной модели, выполняющей задачу минимизации потерь двух моделей в ходе противостояния в минимакс игре. На рис. 3 изображен график изменения функции потерь двух моделей в процессе обучения.

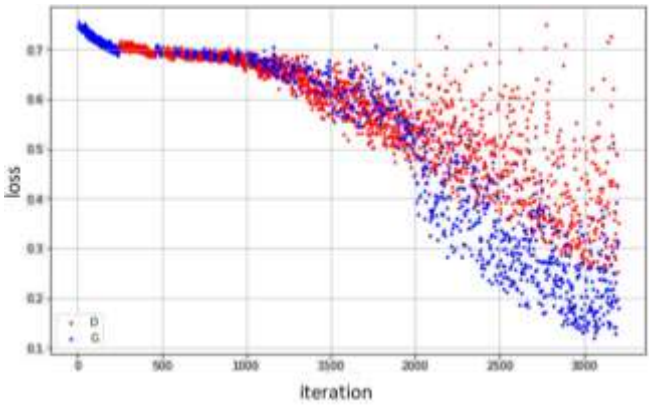

Рис. 3. График изменения потерь моделей в ходе обучения генеративно-состязательным подходом

На графиках видны некоторые проблемы. На рис. 3 обозначены потери моделей во время обучения, красным цветом – потери дискриминатора, синим – потери генератора. График потерь демонстрирует работоспособность состязательного подхода, так как в ходе обучения средняя величина потерь уменьшается. В качестве проблемы на графике можно выделить возрастающий разброс значений потерь в ходе обучения, связано это с тем, что две модели постоянно стремятся переиграть друг друга, и незначительные изменения в параметрах одной модели приводят к значительному ухудшению потерь другой. Подобное поведение может влиять на качество генерируемых данных и приводит к тому, что модель генератора не стремится приблизить вероятностное распределение генератора p_g к распределению реальных данных p_{data}, а вместо этого пытается минимизировать ошибку путём поиска других уязвимостей в модели дискриминатора.

Во время обучения модель обрабатывает каждый пакет данных и считает точность, с которой отработала модель на каждом пакете и для каждой модели. Таким образом, во время обучения происходит вычисление точности моделей. Результаты представлены на рис. 4. По изображению видно, что в некоторый момент у моделей происходит сходимость по точности к значению 1.0, что подтверждает выводы, сделанные к рис. 3. Модель генератора подстраивается под уязвимость дискриминатора и корректирует свои веса таким образом, чтобы минимизировать ошибку.

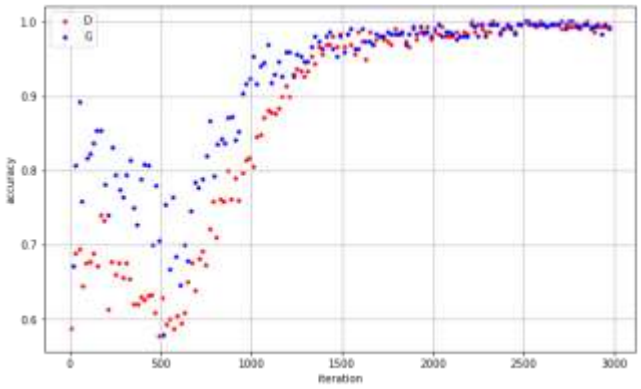

Рис. 4. Точность моделей на каждом шаге обучения

Оценка качества генерации парафраз является отдельным вопросом, и до сих пор не существует очевидных способов измерения качества сгенерированных данных [9]. В данной работе был осуществлен ручной анализ. Примеры текстов представлены на рис. 5. Результаты соответствуют ~600 итерациям обучения, что равняется обучению на 6000 пар текстов.

Исходный текст_____: Ученые Северо-Западного университета (США) открыли метод эффективного лечения диабета первого типа с помощью иммуномодуляции
Парафраза обучающая: Ученые Северо-Западного университета США метод перспективного излечения диабета первого типа с помощью вакцинации открыли
Парафраза GAN_____: Ученые США Северо-Западного университета открыли метод лечения лечения диабета первого типа с помощью открыли ли

Исходный текст_____: Хакеры с помощью специальной техники могут обесточить крупные сети и оставить без электричества и света целый город.
Парафраза обучающая: Хакеры могут крупные сети с помощью специальной техники обесточить и оставить целый город без энергии и света.
Парафраза GAN_____: Хакеры с помощью специальной техники обесточить крупные сети и оставить без электричества и света целый город город

Рис. 5. Пример результатов обучения модели. «Paraphrase_train» – парафраза, использованная для обучения модели. «Paraphrase_GAN» – парафраза, полученная обученной моделью

Можно предположить, что модели удаётся выучить некоторые шаблоны обучающего примера, но она допускает ошибки. При этом, если продолжить обучение, то её результаты начинают значительно ухудшаться, и качество сгенерированных текстов значительно падает.

VI. Заключение

В данной работе было дано описание проблемы применения генеративно-состязательного обучения к обработке текстовых данных, дано описание и обоснована необходимость решения задачи перефразирования текстовых данных, а также было выполнено первичное исследование алгоритма генеративно-состязательного обучения для решения задачи перефразирования текстовых данных. Несмотря на то, что качество обучения пока не является удовлетворительным для подобного рода задачи, построенный алгоритм может иметь потенциал для дальнейших исследований.

В качестве дальнейшего развития работы необходимо более детально провести исследования способов устранения проблемы корреляции признаков, не связанных друг с другом в реальном мире, из-за которой качество генерации текстов значительно падает после продолжительного обучения, а также добавить ряд метрик качества модели перефразирования, несмотря на существующие сложности в данном вопросе.

Список литературы

[1] Goodfellow I. et al. Generative adversarial nets //Advances in neural information processing systems. 2014. Т. 27.

[2] X. Wu, K. Xu and P. Hall. A survey of image synthesis and editing with generative adversarial networks // Tsinghua Science and Technology. Vol. 22. № 6. P. 660-674. December 2017, doi: 10.23919/TST.2017.8195348.

[3] Wang X. et al. GAN-generated Faces Detection: A Survey and New Perspectives //arXiv preprint arXiv:2202.07145. 2022.

[4] De Rosa, G. H., and Papa, J. P. A survey on text generation using generative adversarial networks. Pattern Recognition, 119, 108098. doi:10.1016/j.patcog.2021.108098 (https://doi.org/10.1016/j.patcog.2021.108098)

[5] Yu L. et al. Seqgan: Sequence generative adversarial nets with policy gradient //Proceedings of the AAAI conference on artificial intelligence. 2017. Т. 31. №. 1.

[6] Ke Wang, Xiaojun Wan. SentiGAN: Generating Sentimental Texts via Mixture Adversarial Networks. Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence. 2018. doi: 10.24963/ijcai.2018/618 (https://doi.org/10.24963/ijcai.2018/618)

[7] Fursov I. et al. Differentiable language model adversarial attacks on categorical sequence classifiers //arXiv preprint arXiv:2006.11078. 2020.

[8] Jang E., Gu S., Poole B. Categorical reparameterization with gumbel-softmax //arXiv preprint arXiv:1611.01144. 2016.

[9] Shen L. et al. Revisiting the Evaluation Metrics of Paraphrase Generation //arXiv preprint arXiv:2202.08479. 2022.

[10] Raffel C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer //arXiv preprint arXiv:1910.10683. 2019.