

ГЕНЕРАТИВНО-ДИСКРИМИНАТИВНАЯ НЕЙРОСЕТЕВАЯ МОДЕЛЬ ДЛЯ ЗАДАЧИ АВТОРСКОГО ПРОФИЛИРОВАНИЯ^{1,2}

© 2020 г. А. Г. Сбоев^{1,2,*}, Р. Б. Рыбка¹, А. В. Грязнов¹, И. А. Молошников¹

¹ Национальный исследовательский центр “Курчатовский институт”, Москва, 123098, Россия

² Национальный исследовательский ядерный университет “МИФИ”, Москва, 115409, Россия

*e-mail: sag111@mail.ru

Поступила в редакцию 07.10.2019 г.

После доработки 11.10.2019 г.

Принята к публикации 15.10.2019 г.

В работе рассматривается генеративно-дискриминативная модель (GAN) применительно к задаче анализа текстовых данных, в частности определения пола автора русскоязычного текста. Разработанный подход относится к классу алгоритмов обучения с частичным привлечением учителя, когда в процессе настройки модели задействуются множества размеченных и неразмеченных примеров. GAN модель реализована в виде глубокой нейронной сети, состоящей из полносвязных, рекуррентных и сверточных слоев. Основу генеративной части GAN модели составляет вариационный автоэнкодер, который кодирует входной пример в пространство скрытых переменных и декодирует из них в исходное представление. При декодировании используется метка класса входного примера: известная в случае размеченного множества или определенная классификатором в случае неразмеченного. Входными данными для модели являются последовательности векторов главных компонент для морфологических признаков слов. Для обеспечения функции восстановления текстов, имеющих более 50 слов, используются принципы работы языковых моделей. Дискриминативная часть настраивается определять, был ли поданный пример сгенерирован автоэнкодером или был взят из исходного множества. Оценка качества GAN модели производилась на множестве примеров корпуса текстов блогов LiveJournal. Показано, что применение генеративно-дискриминативной модели позволяет улучшить качество классификации на 2% в F1 метрике, одновременно снижая в 2–3 раза погрешность при обучении на малом количестве размеченных примеров. Исследованы различные режимы обучения и вариации топологии GAN модели. Продемонстрированы наиболее эффективные режимы работы моделей данного типа для задачи классификации текстов.

Ключевые слова: машинное обучение, искусственные нейронные сети, обработка естественного языка, автоматизированный анализ текстов, генеративно-дискриминативные нейронные сети, авторское профилирование, определение пола автора текста

DOI: 10.1134/S2304487X19060129

1. ВВЕДЕНИЕ

Одним из развивающихся направлений в создании классификационных моделей, основанных на глубоких нейронных сетях, являются методы обучения с частичным привлечением учителя (в лит. Semi-supervised). Это обуславливается наличием и постоянным ростом в современных условиях неразмеченных и неструктурированных данных. В то время как развитие успешных топологий глубоких сетей требует наличия большого числа размеченных примеров, получение кото-

рых порой затруднительно и связано с дорогостоящим процессом их разметки. Примером класса подобных задач, где это необходимо, является область авторского профилирования, когда требуется автоматическими средствами определить такие признаки автора как пол, возраст и разные психологические особенности. Процесс разметки при этом состоит в сборе набора текстовых данных респондентов в условиях, когда можно точно проверить искомые признаки, что является трудоемкой задачей. Методы semi-supervised обучения достаточно разнообразны, но идейно сводятся к формированию дополнительного набора примеров, позволяющего улучшить эффективность существующей классификационной модели. Можно выделить 2 основных подхода создания такого множества: 1) генерация новых при-

¹ Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-29-10084 “мк”.

² Работа была выполнена с использованием оборудования центра коллективного пользования “Комплекс моделирования и обработки данных исследовательских установок мега-класса” НИЦ “Курчатовский институт”, <http://ckp.nrcki.ru/>.

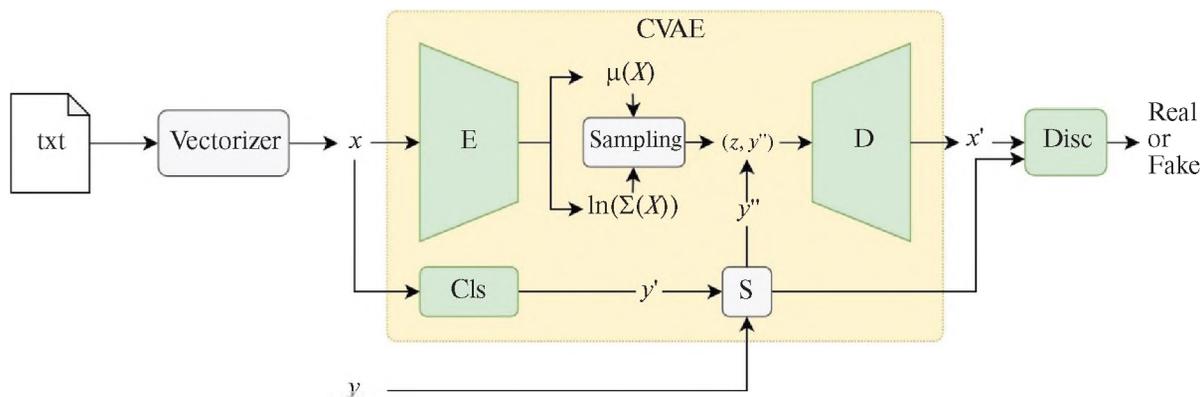


Рис. 1. Схема генеративно-дискриминативной модели GAN на основе условного вариационного автоэнкодера (CVAE) для классификации текста. Vectoriser – векторизатор для кодирования текстовых признаков, E кодировщик, D декодировщик, Cls классификатор, Disc дискриминатор.

меров, 2) использование примеров из числа неразмеченных. Для первого применяются методы генерации текстов на основе специальных топологий нейронных сетей LeakGAN [1], SeqGAN, RankGAN, RelGAN [2], GPT-2 [3] и других, которые обладают значительной сложностью и содержат более 100 млн настраиваемых параметров. Для их обучения необходимы большие массивы данных, близких по специфике к имеющемуся размеченному множеству. Менее ресурсоемки методы второго типа. К ним относятся методы предварительной автоматической разметки большого массива данных несколькими независимыми слабыми моделями с последующим выбором примеров с согласованной оценкой (trigger и др.), вариационные автоэнкодеры, позволяющие кодировать входные примеры в внутреннее пространство и декодировать из него в исходное представление. Добавление в такую архитектуру классификатора позволяет учитывать вклад от неразмеченных примеров в рамках непрерывного обучения классификационной модели [4] и др. Их недостатком является возможность работы только с небольшими размерами текстов, в среднем около 50 слов [5, 6]. На практике для увеличения размеров анализируемых текстов при использовании условного вариационного автоэнкодера (CVAE) применяются специальные алгоритмы, такие как: последовательное декодирование вектора документа в вектора предложений и далее в вектора отдельных слов [7], или подача информации о входной последовательности, например, предыдущие слова.

В данной работе на примере задачи установления пола автора текста мы исследуем метод, основанный на сочетании нескольких подходов в рамках GAN архитектуры (см. рис. 1).

В качестве генеративной части применяется архитектура CVAE с включением механизмов

обучения языковых моделей. Дискриминативная часть в виде сети на базе GRU (General Recurrent Unit) [8] и полносвязных слоев используется для бинарной классификации истинности входного примера. Такая комбинация методов объединяет оба подхода и дает возможность в рамках единой модели проводить обучение классификатора с использованием размеченной и неразмеченной частей данных. Кроме того, использование модульной структуры позволяет более детально контролировать обучение единой модели, за счет синхронизации настройки ее отдельных компонент.

Далее в разделе 2 изложены основные математические предпосылки для построения генеративной части модели. В разделе 3 описаны используемые конфигурации составных частей в рамках единой модели. В разделе 4 представлены описание данных и экспериментальные результаты.

2. МЕТОДЫ И ПОДХОДЫ (МАТЕМАТИЧЕСКИЕ ПРЕДПОСЫЛКИ)

2.1. Условный вариационный автоэнкодер

Условный вариационный автоэнкодер (CVAE) объединяет классификатор и вариационный автоэнкодер, обучаемые совместно. Вариационный автоэнкодер (VAE) состоит из кодировщика и декодировщика. Первый из них осуществляет процесс отображения входного текста на пространство вещественных векторов, а второй позволяет осуществить генеративный процесс. В общем случае такой процесс описывается как:

$$P(X) = \int_Z P(X|Z)P(Z)dZ$$
, где X – это набор примеров исходных данных, Z – это скрытые переменные. Скрытые переменные для каждого X определяются из распределения $Q(Z|X)$. Считается, что модель автоэнкодера настроена верно, ес-

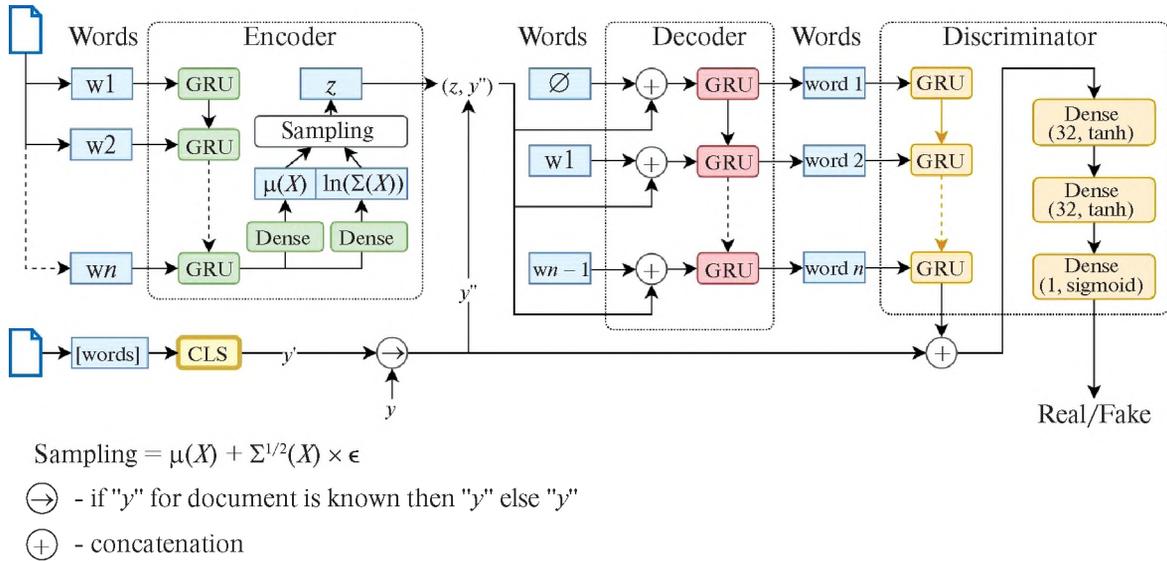


Рис. 2. Конфигурация Кодировщика, Декодировщика и Дискриминатора в составе единой GAN модели, блок “+” означает конкатенацию векторов, блок “→” – блок возвращает реальный y, если он задан (часть размеченных примеров), и предсказанный классификатором y’ – в случае использования неразмеченных примеров. ∅ нулевой вектор, w₁ w_n, [words] слова документа, входы сети, word 1-word n выход CVAE и вход для Дискриминатора.

ли подобранное распределение $Q(Z|X)$ и реальное распределение $P(Z|X)$ близки. Критерием близости этих распределений является дивергенция Кульбака-Лейблера. Применительно к VAE, задачей кодировщика является представление входного текста в пространстве скрытых переменных Z, распределенных по нормальному закону. Таким образом, кодировщик отражает функцию $Q(Z|X, W_Q)$. W_Q – параметры кодировщика.

Задачей декодировщика является восстановление примера из Z обратно в X, т.е. декодировщик выполняет функцию $P(X|Z, W_P)$. W_P – параметры декодировщика.

Основным отличием CVAE от VAE является добавление классификационной модели (классификатора) в состав генератора таким образом, чтобы учесть вклад восстановительной ошибки в процессе обучения классификатора. При этом функция декодировщика изменяется на $P(X|Z, W_P, Y)$, где Y – метка класса текста, которая зависит от входного набора примеров: Y известен в случае размеченного множества, а в случае неразмеченного множества Y определяется классификатором. Таким образом, конфигурация CVAE позволяет использовать для обучения классификатора неразмеченное множество примеров.

2.2. Генеративно-дискриминативная модель

CVAE в составе генеративно-дискриминативной модели обеспечивает генеративную функцию в составе единой GAN модели, а задачей дискриминатора является определение того, восстановлен ли входной пример генератором (Fake)

или он является реальным текстом из выборки (Real).

Модель классификатора состоит из двух сверточных слоев, обрабатывающих входной текст как последовательность закодированных слов. Каждый состоит из 64 нейронов с функцией активации ReLU: $f(x) = \max(x, 0)$, обрабатывающих входной массив с окном 4. После каждого слоя выполняется операция MaxPooling1D с окном 3. Далее для всего текста формируется один вектор при помощи операции GlobalMaxPooling1D. Полученный вектор обрабатывается последовательно двумя полносвязными слоями. Первый содержит 32 нейрона с сигмоидальной функцией активации: $f(x) = 1/[1 + \exp(-x)]$, а второй – 2 нейрона с функцией активации softmax: $f(x)_i = e^{x_i} / \sum_k e^{x_k}$ (где i и k – индексы нейронов слоя, а x значение их активации) отвечающих за класс пола автора в решаемой классификационной задаче.

Конфигурация кодировщика, декодировщика и дискриминатора основана на рекуррентных GRU и полносвязных (Dense) слоях (см. рис. 2).

Кодировщик. После обработки входной последовательности GRU слоем кодировщика последнее скрытое состояние подается в качестве входа двум полносвязным слоям для получения векторов средних значений и логарифмов дисперсий $\mu(X)$ и $\ln(\Sigma(X))$. Далее формируется код документа с использованием функции сэмплирования (sampling): $z = \mu(X) + \sqrt{\Sigma(X)}\epsilon$, где $\epsilon \in N(0, 1)$ – случай-

ный вектор шума выбранный из нормального распределения.

Полученный код документа z конкатенируется с эталонной меткой класса документа, если она известна, в противном случае с выходом классификатора y' .

Декодировщик. На этапе восстановления примера с использованием декодировщика, полученный вектор $\langle z, y \rangle$ копируется N раз, где N – количество слов во входном примере. Вход для декодировщика формируется путем добавления каждой i -й паре $\langle z, y \rangle$ кода $i - 1$ слова входного примера, т.е. X_{i-1} .

Выходом декодировщика, состоящего из слоя GRU, является восстановленный входной пример X' .

Дискриминатор. Дискриминатор состоит из последовательности GRU слоев, двух полносвязных слоев по 32 нейрона с функцией активации \tanh , и 1 одного полносвязного слоя с одним нейроном и функцией активации sigmoid . По активности нейрона последнего слоя определяется класс входного примера. Входом для дискриминатора является набор примеров, как исходных из размеченного множества, так и восстановленных из неразмеченного множества с использованием генеративной части. Предварительно примеры конкатенируются с метками классов, взятыми либо из тренировочного набора, либо полученными из классификатора. Для реализации и настройки модели используются библиотеки Keras [9] 2.2.4 и Tensorflow [10] 1.12.

2.3. Процесс обучения

Процесс обучения всей модели выполняется по этапам. На первом этапе учится классификатор на доступном размеченном множестве, обучение производится до 1000 эпох с контролем переобучения по валидационному множеству с применением метода раннего останова. При росте ошибки на валидационном множестве в течение 30 эпох производится остановка обучения и загрузка лучших весов.

При обучении классификатора используется функция потерь категориальная кросс-энтропия (categorical_crossentropy) [11], и оптимизатор Adam [12] (параметры по умолчанию).

На втором этапе проводится обучение генеративной части: CVAE. В качестве функции потерь используется комбинация из:

1. функции потерь классификатора (cls_loss), рассчитываемой на примерах с известной меткой класса y и предсказанных метках y' . Для расчета используется функция средней квадратичной ошибки (MSE): $\text{cls_loss} = \text{MSE}(y, y')$;

2. дивергенции Кульбака-Лейблера (KL_loss), рассчитываемого на основе близости распределе-

ния скрытых переменных, полученных после работы Кодировщика, к нормальному распределению по формуле:

$$\text{KL_loss} = -0.5 \sum_{k=1}^{256} [1 + \ln(\Sigma(X)_k) - \mu(X)_k^2 - \Sigma(X)_k],$$

где k – индекс компоненты вектора Z .

3. функции потерь при декодировании примеров (D_loss), рассчитываемой для всех примеров функцией MSE: потерь восстановления $\text{D_loss} = \text{MSE}(X, X')$, где X – набор входных примеров, а X' – набор восстановленных примеров.

Для обучения различных компонент CVAE части используются комбинации описанных выше функций. Веса декодировщика изменяются под действием D_loss , для настройки весов классификатора используется сумма функций потерь 1 для размеченной части данных и 3 для неразмеченной. Для настройки весов кодировщика применяется сумма функций потерь 2 и 3.

При обучении используется контроль переобучения по суммарной функции потерь $\text{CVAE_loss} = \text{cls_loss} + (\text{KL_loss} + \text{D_loss})/2$ на валидационном множестве с ранним остановом, аналогично классификатору. После раннего останова производится дообучение классификатора на размеченном множестве (аналогично этапу 1).

На третьем этапе производится обучение всей модели GAN, которая состоит из генеративной части на основе CVAE и дискриминативной на базе GRU слоев. Задача дискриминатора – отличить восстановленные примеры (метка fake) от оригинальных (метка real) (см. рис. 1). Задача генератора – научиться генерировать примеры так, чтобы дискриминатор ошибался. Обучение GAN длится до 50 эпох, при этом на одно обновление весов при обучении дискриминатора (режим обучения: “пакетный”) приходится две итерации обучения генератора. По прошествии 50 эпох классификатор дообучается на размеченном множестве с параметрами обучения с этапа 1.

3. ЭКСПЕРИМЕНТЫ

3.1. Корпус размеченных примеров

Корпус содержит 21194 текстов от 2694 авторов LiveJournal [13]. Блоги отбирались вручную на основе анализа текстов экспертами и указанных данных авторов: пола и имени. Для предобработки текстов, токенизации по словам и отдельным предложениям, а также для морфологического разбора использовалась библиотека UDPipe [14].

Для экспериментов была выбрана часть текстов корпуса длиной менее 250 слов (далее корпус). Выбранный корпус содержит 3967 текстов

от 1872 авторов LiveJournal, средняя длина текста: 13 предложений, 214 слов. Соотношение текстов от авторов разного пола: 1429 от женщин и 2538 от мужчин.

3.2. Кодирование текстов

Метка класса текста, т.е. пола автора, представляется в виде унитарного кода [1, 0] – женщины, [0, 1] – мужчины. Последовательности слов в текстах кодируются только комбинацией морфологических признаков (тегов) без учета словоформ и лемм для создания контекстно-независимой модели классификатора.

Для корректной работы генеративной части GAN модели необходимо представить вектора признаков слов в непрерывном пространстве. Для этого набор морфологических тэгов кодируется бинарно, размерность вектора равна всем встречающимся значениям морфологических признаков в тренировочном множестве. Если признак присутствует в векторе морфологических признаков, то в позиции, соответствующей этому признаку в векторе, ставится 1, если нет – 0. Далее для перехода к непрерывному пространству из бинарного производится преобразование полученных бинарных векторов методом главных компонент [15] до 32 главных компонент. Вектор главных компонент используется в качестве входных данных для GAN модели.

3.3. Разделение корпуса на подмножества

Для экспериментов проводилась балансировка корпуса по классам путем удаления текстов более представительного класса. После балансировки проводилось разбиение на несколько множеств:

- размеченное (тренировочное) – содержит метки пола для документа;

- неразмеченное содержит тексты без указания пола автора;

- валидационное содержит тексты с указанием пола автора и используется для раннего останова;

- тестовое содержит метки пола и используется для оценки точности модели на разных этапах обучения.

Разбиение на размеченное, неразмеченное, валидационное и тестовые множества делалось по текстам без учета идентификатора автора текста. Во всех экспериментах на выбранном корпусе примеров размер валидационного и тестового множеств равен 448 и 544, соответственно. Размеры размеченного множества изменялись в зависимости от эксперимента. Для тренировочного множества использовались 3 набора по 128, 224, 416 текстов.

Неразмеченное множество выбиралось исходя из предположения, что оно должно отражать генеральную совокупность для текущей задачи.

4. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Предварительно был проведен ряд экспериментов для составления неразмеченного множества примеров. Для этого были построены графики зависимости точности классификатора от размера тренировочных данных для двух случаев. На рис. 3 показаны результаты экспериментов при выборе примеров для тестировочной, валидационной и тренировочной частей случайно. На экспериментах, результаты которых отражены на рис. 4, наборы данных для тестирования и валидации фиксировались, а к тренировочному множеству постепенно добавлялись новые данные примеры. На каждый размер обучающего набора

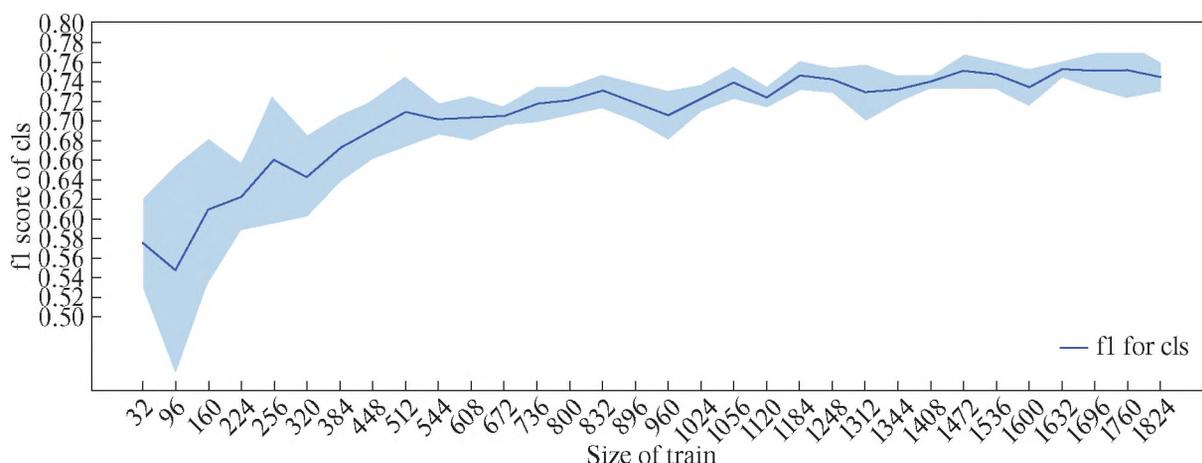


Рис. 3. Зависимость точности классификатора от количества размеченных данных при выборе случайных наборов примеров для обучения, валидации и тестирования.

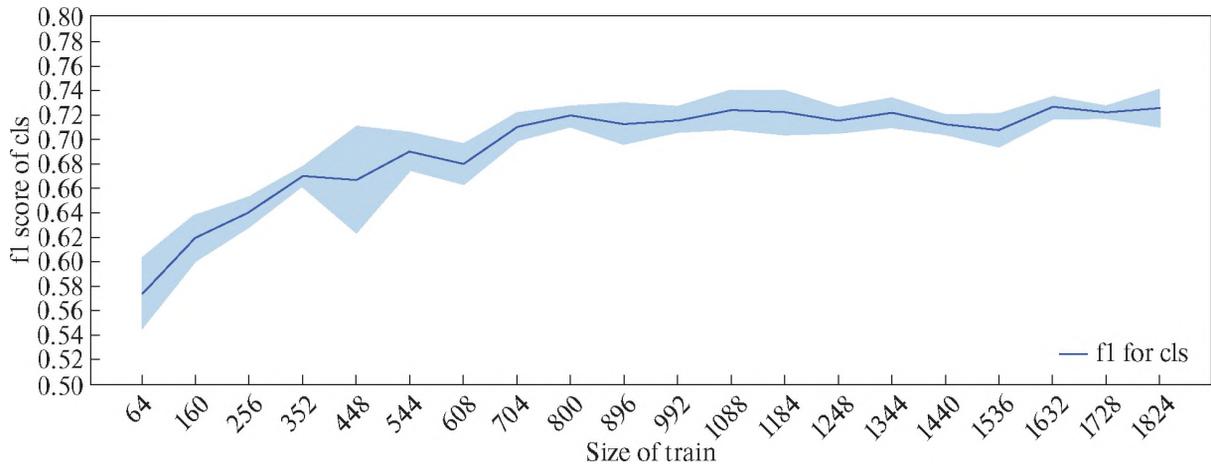


Рис. 4. Зависимость точности классификатора от количества размеченных данных при зафиксированных валидационном и тестировочном множествах и с постепенным увеличением тренировочного набора.

было проведено по 5 запусков для усреднения результатов и оценки отклонения от среднего.

В результате установлено, что точность решения классификационной задачи не увеличивается при использовании набора тренировочных примеров размером более 1000 текстов. Таким образом, подмножество такого объема достаточно для описания задачи и будет использоваться для проведения дальнейших экспериментов в качестве неразмеченного множества.

На рис. 5 показана точность классификатора после различных этапов обучения GAN модели и в зависимости от размера тренировочного множества. При этом состав и размеры валидационного и тестового множеств не изменялись, а состав тренировочного изменялся от запуска к запуску случайно. Каждый эксперимент для заданного размера тренировочного множества запускался 30 раз.

Из графика следует, что обучение классификатора в составе генеративной части на базе CVAE повышает среднюю точность классификатора на 2%. Более значимым является эффект уменьшения разброса в результатах при множественных запусках (кросс валидации). Как следует из рис. 4, стандартное отклонение при маленьких тренировочных выборках высокое и составляет порядка 5–6%. В то время как использование обучения на неразмеченном множестве снижает этот разброс в 2–3 раза и делает его приемлемым. Использование дискриминатора также позволяет улучшить точность классификатора после обучения CVAE при небольшом наборе тренировочных примеров. Это подтверждает рис. 5, где показаны точности настройки сети с усреднением по нескольким запускам обучения на независимых разбиениях по всей совокупности размеченных примеров, а также с фиксированными валидационным и тестировочным множествами.

5. ЗАКЛЮЧЕНИЕ

Результаты проведенных экспериментов показывают, что при небольших размерах размеченных текстовых выборок при наличии достаточной небольшой выборки можно получать, с одной стороны, более точные результаты, а с другой стороны, с меньшей погрешностью. В плане про-

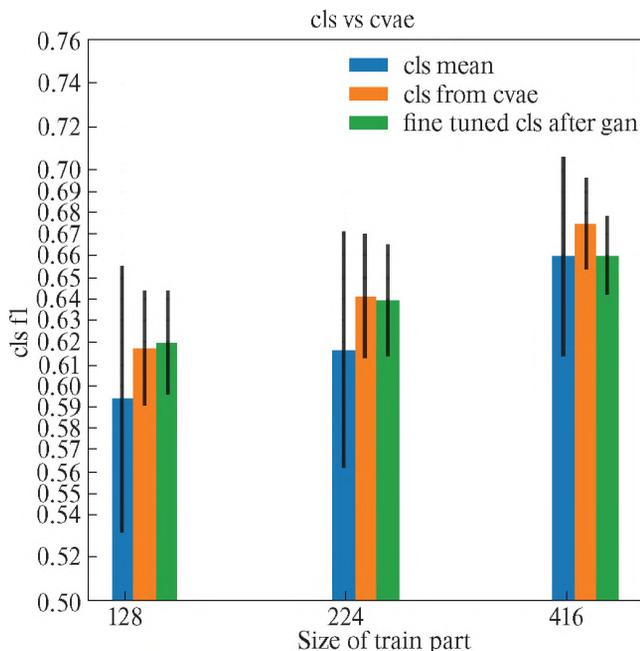


Рис. 5. Результаты обучения классификатора и дообучения с CVAE и GAN при зафиксированных тестовом, валидационном и неразмеченном наборах.

ведения дальнейших работ надо отметить, что требует исследования оптимальность выбора в качестве распределения латентных признаков для текстовых задач нормального распределения.

СПИСОК ЛИТЕРАТУРЫ

1. Guo J. et al. Long text generation via adversarial training with leaked information // Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
2. Nie W., Narodytska N., Patel A. RelGAN: Relational generative adversarial networks for text generation. 2018.
3. Radford A. et al. Language models are unsupervised multitask learners // OpenAI Blog. 2019. Т. 1. № 8.
4. Bao J. et al. CVAE-GAN: fine-grained image generation through asymmetric training // Proceedings of the IEEE International Conference on Computer Vision. 2017. С. 2745–2754.
5. Xu W. et al. Variational autoencoder for semi-supervised text classification // Thirty-First AAAI Conference on Artificial Intelligence. 2017.
6. Semeniuta S., Severyn A., Barth E. A hybrid convolutional variational autoencoder for text generation // arXiv preprint arXiv:1702.02390. 2017.
7. Shen D. et al. Hierarchically-Structured Variational Autoencoders for Long Text Generation. 2018.
8. Chung J. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling // arXiv preprint arXiv:1412.3555. 2014.
9. Chollet F. et al. Keras. 2015.
10. Abadi M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems // arXiv preprint arXiv:1603.04467. 2016.
11. Arjovsky M., Shah A., Bengio Y. Unitary evolution recurrent neural networks // International Conference on Machine Learning. 2016. С. 1120–1128.
12. Kingma D.P., Ba J. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. 2014.
13. Litvinova T.A., Sboev A.G., Panicheva P.V. Profiling the Age of Russian Bloggers // Conference on Artificial Intelligence and Natural Language. Springer, Cham, 2018. С. 167–177.
14. Straka M., Hajic J., Straková J. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing // Proceedings of the tenth international conference on language resources and evaluation (LREC 2016). 2016. С. 4290–4297.
15. Tipping M.E., Bishop C.M. Probabilistic principal component analysis // Journal of the Royal Statistical Society: Series B (Statistical Methodology).

Vestnik Natsional'nogo issledovatel'skogo yadernogo universiteta "MIFI", 2020, vol. 9, no. 1, pp. 50–57

Generative-Discriminative Neural Network Model for the Task of Author Profiling^{3,4}

A. G. Sboev^{a,b,#}, R. B. Rybka^a, A. V. Gryaznov^a, and I. A. Moloshnikov^a

^a National Research Center "Kurchatov Institute", Moscow, 123098, Russia

^b National Research Nuclear University "MEPhI" (Moscow Engineering Physics Institute), Moscow, 115409, Russia

[#]e-mail: sag111@mail.ru

Received October 7, 2019; revised October 11, 2019; accepted October 15, 2019

Abstract—The paper considers the generative-discriminative model (GAN) as applied to the task of analyzing text data, in particular, determining the gender of the author of a Russian-language text. The approach developed belongs to semi-supervised learning algorithms, when both labeled and unlabeled samples are involved in the model fitting process. The GAN model is implemented as a deep neural network consisting of fully connected, recurrent and convolution layers. The basis of the generative part of the GAN model is a variational auto-encoder, which encodes the input sample into the space of hidden variables and then decodes the latter into the original representation. When decoding, the class label of the input example is used, known in the case of the labeled set or predicted by the classifier for unlabeled samples. The input for the model is a sequence of words, each encoded by a vector of the principal components of its morphological features. To provide the function of recovering texts with more than 50 words, the principles of the work of language models are used. The discriminant part is configured to determine whether a given sample was generated by an auto-encoder or taken from the original set. Quality assessment of the GAN model was carried out on a set of texts from LiveJournal blogs. It is shown that the use of the generative-discriminative model allows to improve the quality of classification by 2% in the F1 metric, while reducing the standard deviation by 2–3 times when learning on a small number of labeled examples. Various modes of training and variations in the topol-

³ The reported study was funded by RFBR according to the research project № 18-29-10084 "МК"

⁴ This work has been carried out using computing resources of the federal collective usage center Complex for Simulation and Data Processing for Mega-science Facilities at NRC "Kurchatov Institute", <http://ckp.nrcki.ru/>.

ogy of the GAN model are investigated, and the most effective modes of operation of models of this type for the task of classifying texts are demonstrated.

Keywords: machine learning, artificial neural networks, natural language processing, automated text analysis, generative-discriminative neural networks, author profiling, author gender identification

DOI: 10.1134/S2304487X19060129

REFERENCES

1. Guo J. et al. Long text generation via adversarial training with leaked information // Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
2. Nie W., Narodytska N., Patel A. RelGAN: Relational generative adversarial networks for text generation. 2018.
3. Radford A. et al. Language models are unsupervised multitask learners // OpenAI Blog. 2019. V. 1. № 8.
4. Bao J. et al. CVAE-GAN: fine-grained image generation through asymmetric training // Proceedings of the IEEE International Conference on Computer Vision. 2017. P. 2745–2754.
5. Xu W. et al. Variational autoencoder for semi-supervised text classification // Thirty-First AAAI Conference on Artificial Intelligence. 2017.
6. Semeniuta S., Severyn A., Barth E. A hybrid convolutional variational autoencoder for text generation // arXiv preprint arXiv:1702.02390. 2017.
7. Shen D. et al. Hierarchically-Structured Variational Autoencoders for Long Text Generation. 2018.
8. Chung J. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling // arXiv preprint arXiv:1412.3555. 2014.
9. Chollet F. et al. Keras. 2015.
10. Abadi M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems // arXiv preprint arXiv:1603.04467. 2016.
11. Arjovsky M., Shah A., Bengio Y. Unitary evolution recurrent neural networks // International Conference on Machine Learning. 2016. P. 1120–1128.
12. Kingma D.P., Ba J. Adam: A method for stochastic optimization // arXiv preprint arXiv:1412.6980. 2014.
13. Litvinova T.A., Sboev A.G., Panicheva P.V. Profiling the Age of Russian Bloggers // Conference on Artificial Intelligence and Natural Language. Springer, Cham, 2018. P. 167–177.
14. Straka M., Hajic J., Straková J. UDPipe: trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, pos tagging and parsing // Proceedings of the tenth international conference on language resources and evaluation (LREC 2016). 2016. P. 4290–4297.
15. Tipping M.E., Bishop C.M. Probabilistic principal component analysis // Journal of the Royal Statistical Society: Series B (Statistical Methodology).