Style Change Detection

Зуева Н., Кузнецова Р., Стрижов В.
Московский Физико-Технический Институт Государственный Университет

2018

1 Abstract

Рассматриваются методы обнаружения мультиавторности в тексте. Исследование сконцентрировано на улучшении качества обнаружения по сравнению с существующими методами. Каждый автор имеет уникальный набор стилистических признаков и изменение стиля распознается как мультиавторство. В данной статье решается задача поиска мультиавторности в том случае, когда нет доступа ко внешним коллекциям.
Ключевые слова: *поиск плагиата, нейронные сети, GAN, обучение с учителем, мультиавторность текста*

2 Введение

Задача поиска *мультиавторства* бывает двух видов: обнаружение внешних заимствований и обнаружение внутренних заимствований.
В первом случае сравнивается подозрительный на мультиавторство текст с коллекцией внешних документов. Для решения задачи анализа внутренних заимствований [3] нужно найти подозрительный текст в отсутствия внешнего корпуса (внешней коллекции). По каждому автору составляется профиль, где указывается стиль письма, пунктуация и прочие признаки, выявленные алгоритмом.
Мы будем распознавать документ как мультиавторный, если не существует *главного автора*, который написал 70Условие «один главный автор» означает следующую общую схему[7] для обнаружения встроенного плагиата [5]:

1. разделение текстового документа на сегменты (например, предложения)
2. разработка набора функций сегмента и объединение их с характеристикой стиля автора, которая измеряет соответствие авторского стиля для каждого текста
3. поиск критических значений в профиле автора для обнаружения плагиата

Конкурс PAN-2018[6] предлагает решить задачу поиска плагиата бинарно: есть ли главный автор у текста. Традиционно задачи обнаружения плагиата решаются при помощи частотного и статистического анализа текста [7], но известны попытки решения задачи поиска мультиавторства и при помощи нейросетей[8]. а также при помощи словесных *n-грамм* — авторы в [11] предложили разделить текстовый документ на набор пересекающихся сегментов (подход «sliding window»).
В данной работе предлагается решать задачу, используя $generative_adversarial_networks$ [2] — генеративная модель порождает тексты в одном авторском стиле, дискриминативная модель — бинарный классификатор. В статье представлены итоги реализации, описание и тестирование алгоритма, который по предположению даст прирост в тестовых показателях для проверки документа на плагиат.

3 Постановка задачи

Поставим задачу формально. Выборка должна быть достаточно большой (свыше 10 000 экземпляров) с документами на английском языке и отдельными файлами, где указаны участки плагиата в каждой статье.

Описание выборки

Для эксперимента используется выборка документов PAN-2018 [8], содержащая коллекцию документов с небольшим числом (до 20) авторов-кандидатов, которые используют заимствования в тексте без указания

авторов другого (неизвестного) набора документов. Известные документы принадлежат нескольким темам, хотя и не обязательно одинаковы для всех авторов-кандидатов.

Предоставляется одинаковое количество документов для каждого автора-кандидата. Неизвестные документы неравномерно распределены по авторам. Длина текста варьируется от 500 до 1000 слов. Документы представлены на английском языке.

Пусть нам дана коллекция текстовых документов D, здесь d_k — каждый отдельный текстовый документ. В пару к d_k ставится в соответствия $t_k \in 0, 1$. Она принимает значение 0, если документ моноавторный и 1, если авторов несколько. Требуется построить алгоритм-классификатор $a : D \to y_k$, который получает на вход документ и проверяет его на плагиат, попутно минимизируя *функцию ошибки* erf:

$$erf = -\frac{1}{|D|} \sum [t_k log y_k + (1 - t_k) log(1 - y_k)$$

$$a = \frac{1}{|D|}] \sum argmin_a erf(y_k, t_k)$$

Критерий качества

Будем использовать те же критерии качества, что и те, которые применяются в соревновании RAN-2018 [8] и [9]. Пусть D это коллекция документов, а каждый ее элемент $--$ d, а s это совокупность некоторых сегментов текста. Рассмотрим пары (s, d), они будут представлять последовательность символов, которая помечена человеком как заимствование в документе d. $S = S = si$ — совокупность всех заимствованных частей текста. За пару (r, d) обозначим последовательность, помеченную алгоритмом как плагиат. $R = ri$ — совокупность всех сегментов, которые нейросеть пометила как заимствованные. Обозначим S_R множество заимствованных частей текста, которые были обнаружены алгоритмом. R_s $--$ части текста, отмеченные сетью, которые находят данную часть заимствований s. Рассмотрим меры качества из PAN-2018 [8]:

1. $Prec(S, R) = \frac{1}{|R|} \sum \frac{|(s_i \cup r_j)|}{|r_j|}$
2. $Rec(S, R) = \frac{1}{|S|} \sum \frac{|(s_i \cup r_j)|}{|s_j|}$

$Precision$ характеризует долю верного распознавания плагиата ко всем выделенным частям документа, $Recall$ характеризует долю правильного распознавания плагиата по отношению ко всем выделенным частям в тексте. Также нам потребуется $F1(S, R)$ мера, которая является отношением произведения $Precision$ и $Recall$ и их суммы: $F1(S, R) = \frac{Prec(S,R) \cdot Rec(S,R}{Prec(S,R} + Rec(S, R)$. Итоговая мера качества P определяется по формуле:

$$P(S, R) = \frac{F1(S,R)}{log_2(1 + \frac{1}{|S_R|} \sum |R_{si}|}$$

Формальная постановка задачи

Пусть у нас есть коллекция документов D и некоторые размеченные документы. Тогда мы можем организовать две выборки $--$ $train$ и $test$. $Train - -$ обучающая выборка с размеченными данными, на которой мы будем обучать нашу GAN[4].

Пусть сеть G это генеративная модель $--$ сеть, генерирующая образцы, а D это дискриминативная модель, сеть, которая будет стараться отличить подлинники от плагиата. Таким образом, алгоритм должен выдать в качестве результата вектор, где будут построчно записаны все заимствования.

4 Базовый эксперимент

В качестве базового алгоритма приведем уже существующий, описанный в статье [8]. Целью этого базового эксперимента ставится проверка того, что заимствованные сегменты текста имеют отличные от среднего вектора значения признаков.

В качестве признака выберем частоты встречаемости слов, т.е. каждому слову ставится в отношение некоторое число, характеризующее частоту встречаемости. Чем больше это число, тем чаще встречается это слово. Обозначим слово за w, тогда в соответствие ему ставится $frec_w = ln\frac{n_w}{n_m ax}$, здесь n_w $--$ число слов w, встреченных в сегменте текста, а n_{max} $--$ число вхождений в тот же сегмент текста самого встречаемого слова. Для проверки данной гипотезы используем критерий $-$. Пусть m_j = среднее значение j-го признака рассматриваемого документа, а r_j это среднеквадратичное отклонение. Пусть $t_{ij} = \frac{x_j - m_j}{r_j}$, здесь tij – нормализованный признак j это i-го сегмента текста. За сегменты ti возьмем предложения из документа. Для каждого предложения t_i строился вектор признаков t_i при помощи технологии $word_to_vec$ и затем подсчитаем отклонение от усредненного по всему тексту вектора t в $L1$- метрике: $\sigma(t_i) = ||t_i - t_{||}$, эксперимент проводится на данных PAN-2016[6].

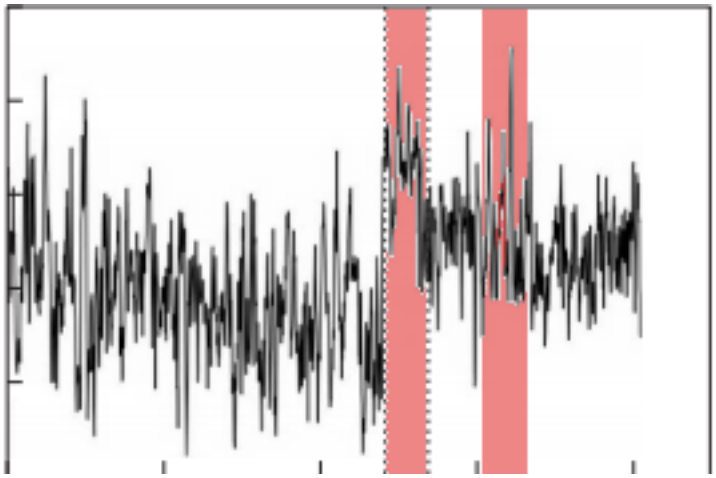

рис.1

На рисунке по оси Ox отложены сегменты текста, а по Oy – значения статистики. Красным отмечены участки, которые были помечены экспертом как заимствованные.

Можно заметить, что заимствованные части документов имеют характерные выбросы из области средних значений x. Но также мы можем наблюдать выбросы там, где их быть не должно, т.е. ошибки первого рода. Выходит, что этот признак работает недостаточно хорошо и надо искать другие, более совершенные методы.Результаты для различных моделей представлены в Таблице 1. Таблица 2 показывает результаты для лучшей модели отдельно по сложениям 6-10 и в среднем. Окончательно достигнутое качество составляет 0,29 для F1-меры и 0,21 для macro-pladget[5].

Table 1: Results for test folds, selecting the best model

			Macro				Micro			
Test	F1-raw	Gran	Rec	Prec	**F1**	**Pladget**	Rec	Prec	F1	Pladget
fold 1	0.43	1.58	0.36	0.23	0.28	0.207	0.48	0.43	0.45	0.329
fold 2	0.41	1.57	0.35	0.23	0.28	0.205	0.45	0.40	0.42	0.311
fold 3	0.36	1.70	0.30	0.20	0.24	0.168	0.41	0.39	0.40	0.278
fold 4	0.45	1.53	0.38	0.28	0.32	**0.242**	0.45	0.46	0.46	0.341
fold 5	0.43	1.62	0.34	0.30	0.32	0.228	0.44	0.51	0.47	0.338

Table 2: Results for validation

			Macro				Micro			
Valid	F1-raw	Gran	Rec	Prec	**F1**	**Pladget**	Rec	Prec	F1	Pladget
fold 6	0.43	1.62	0.39	0.22	0.28	0.203	0.50	0.40	0.45	0.320
fold 7	0.45	1.73	0.39	0.25	0.31	0.213	0.48	0.46	0.47	0.323
fold 8	0.41	1.56	0.37	0.22	0.28	0.203	0.48	0.41	0.44	0.326
fold 9	0.43	1.69	0.37	0.26	0.31	0.216	0.44	0.43	0.43	0.303
fold 10	0.36	1.48	0.33	0.19	0.24	0.186	0.43	0.34	0.38	0.290
mean	**0.42**	**1.62**	**0.37**	**0.23**	**0.29**	**0.206**	**0.47**	**0.41**	**0.44**	**0.315**

5 Алгоритм

GAN способны моделировать сложные многомерные распределения реальных данных, что предполагает их эффективность для задачи обнаружения аномалий. Однако в немногих работах было изучено использование GAN для задачи обнаружения аномалий. Мы используем недавно разработанные модели GAN для обнаружения аномалий.

1. Сегментирование текста.
   Исходный текст подвергается предобработке: удаляются служебные символы, все буквы перево- дятся в нижний регистр. Также из текста удаляются стоп-слова. Текст разбивается на предложения. Затем формируется разбиение текста на сегменты t_i : если длина очередного предложе- ния меньше минимальной длины сегмента $lsegm$, к этому предложению добавляется следующее за ним — процесс повторяется, пока длина сегмен- та t_i не превысит заданную минимальную длину. Минимальная длина сегмента $lsegm$ является настраиваемым параметром алгоритма.

2. GAN модель.
   Пусть X это пространство объектов, в нашем случае это статьи из соревнования PAN-2018. На некотором вероятностном пространстве Ω задана векторная случайная величина $x : \Omega \to X$, с распределением вероятностей, имеющим плотность $p(x)$, такую, что подмножество пространства X, на котором $p(x)$ принимает ненулевые значения — это тексты, где $t_k = 1$. Пусть имеется выборка вида $[x_i, i \in [1, N], x_i\ p(x)]$.
   Аналогично определим вспомогательное пространство Z и случайную величину $z : \Phi \to Z$ с распределением вероятностей, имеющим плотность $q(z)$.
   $D : X \to (0, 1)$ — функция-дискриминатор. Эта функция принимает на вход объект $x \in X$ (текст некоторого размера) и возвращает вероятность того, что входной текст является мультиавторным. $G : Z \to X$ — функция-генератор. Она принимает значение $z \in Z$ и выдает объект пространства X, то есть текст. Предположим, что у нас уже есть идеальный дискриминатор D. Для любого примера x он выдает истинную вероятность принадлежности этого примера заданному подмножеству X, из которого получена выборка $x_i, i = 1..N$. Переформулируя задачу обмана дискриминатора на вероятностном языке мы получаем, что необходимо *максимизировать* вероятность, выдаваемую идеальным дискриминатором на сгенерированных примерах. Таким образом оптимальный генератор находится как

$$G* = argmax_g E_{z\ q(x)} D_k(G(z))$$

. Известно, что $log(x)$ монотонно возрастает и не меняет положения экстремумов аргумента, то эту формулу переписать в виде:

$$G* = argmax_g E_{z\ q(x)} log D_k(G(z)),$$

что будет удобно далее.

В реальности идеального дискриминатора нет. Так как задача дискриминатора — предоставлять сигнал для обучения генератора, вместо идеального дискриминатора достаточно взять дискриминатор, *идеально отделяющий настоящие примеры от сгенерированных текущим генератором*, т.е. идеальный только на подмножестве X из которого генерируются примеры текущим генератором.

Эту задачу можно переформулировать, как поиск такой функции D, которая максимизирует вероятность правильной классификации примеров как настоящих или сгенерированных. Это называется задачей *бинарной классификации*[14] и в данном случае мы имеем бесконечную обучающую выборку: конечное число настоящих примеров и потенциально бесконечное число сгенерированных примеров. У каждого примера есть метка: настоящий он или сгенерированный, в наших обозначениях это t_i.

Воспользуемся методом максимального правдоподобия:[14]

Выборка:

$$S = [(x, 1), x\ p(x)] \cup [(G(z), 0), z\ q(z)]$$

Определим плотность распределения $f(\xi|\eta = 1) = D(\xi), f(\xi|\eta = 0) = 1 - D(\xi)$, тогда $f(\xi|\eta)$ — суть дискриминатор D, выдающий вероятность класса 1 (мультиавторство в тексте) в виде распределения на классах $0, 1$. Так как $D() \in (0, 1)$, это определение задает корректную плотность вероятности. Тогда оптимальный дискриминатор можно найти как:

$D* = f*(\xi|\eta) = argmax_f f(\xi_1, ...|\eta_1, ... = argmax_f \prod f(\xi_i|\eta_i)$ Знаем, что η_i принимает значения $0, 1$, тогда:

$$D* = argmax_f \prod_{i,\eta=1} f(\xi_i|\eta = 1) \prod_{i,\eta=0} f(\xi_i|\eta = 0) =$$
$$= argmax_D [\sum_{x_i\ p(x)} log D(x_i) + \sum_{z_i\ q(z)} log(1 - D(G(z_i)))]$$

Устремим размер выборки в бесконечность:
$D* = argmax_D E_{x_i\ p(x_i)} log(D(x_i) + E_{z_i\ q(z) log(1-D(G(z_i)))}$
Итак, $D*$ это оптимальный дескриминатор.

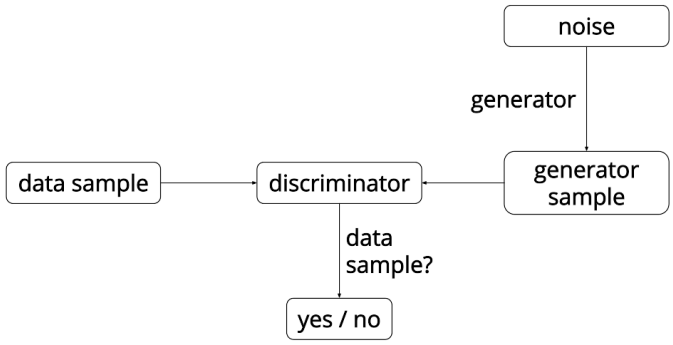

рис.2

На рис.2 представлена схема работы описанного выше процесса. *решающий алгоритм*:

(a) Устанавливаем некоторый начальный $G_0(z)$

(b) Начинается k-я итерация, $k = 1...K$

(c) Ищем оптимальный для текущего генератора дискриминатор:
$D_k = argmax_D E_{x_i\ p(x)} log D(x_i) + E_{z_i\ q(z)} log(1 - D)k - 1(G(z_i)))$

(d) Улучшаем дискриминатор, используя оптимальный дискриминатор:
$G_k = argmax_G E_{z\ q(x)} log(1 - D(G_{k-1(z_i)}))$ Важно находиться в окрестности текущего генератора. Если отойти далеко от текущего генератора, то дискриминатор перестанет быть оптимальным и алгоритм перестанет быть верным.

6 Вычислительный эксперимент

1. Данные загружаются из соревнования PAN-2018
2. Данные полные, с хорошей разметкой, поэтому особой предобработки выполнять не будем.
3. На графиках представлен процесс обучения:

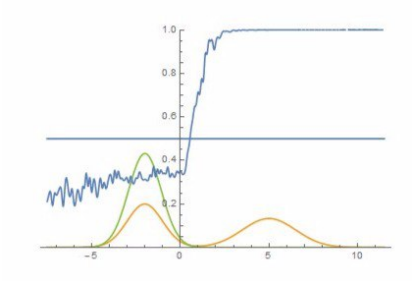

рис.3 Начало обучения

1. В результате выполнения алгоритма было получен:
dLossReal: 0.6914341 dLossFake: 0.71694636
2. В результате получили, что на данной коллекции использование generative-advertisal-network дает незначительный прирост в качестве.

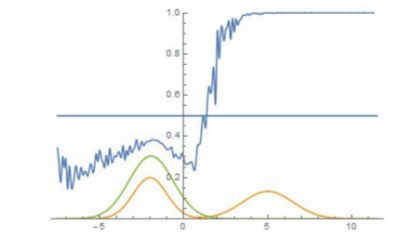

рис.4 Конец обучения

7 Анализ результатов

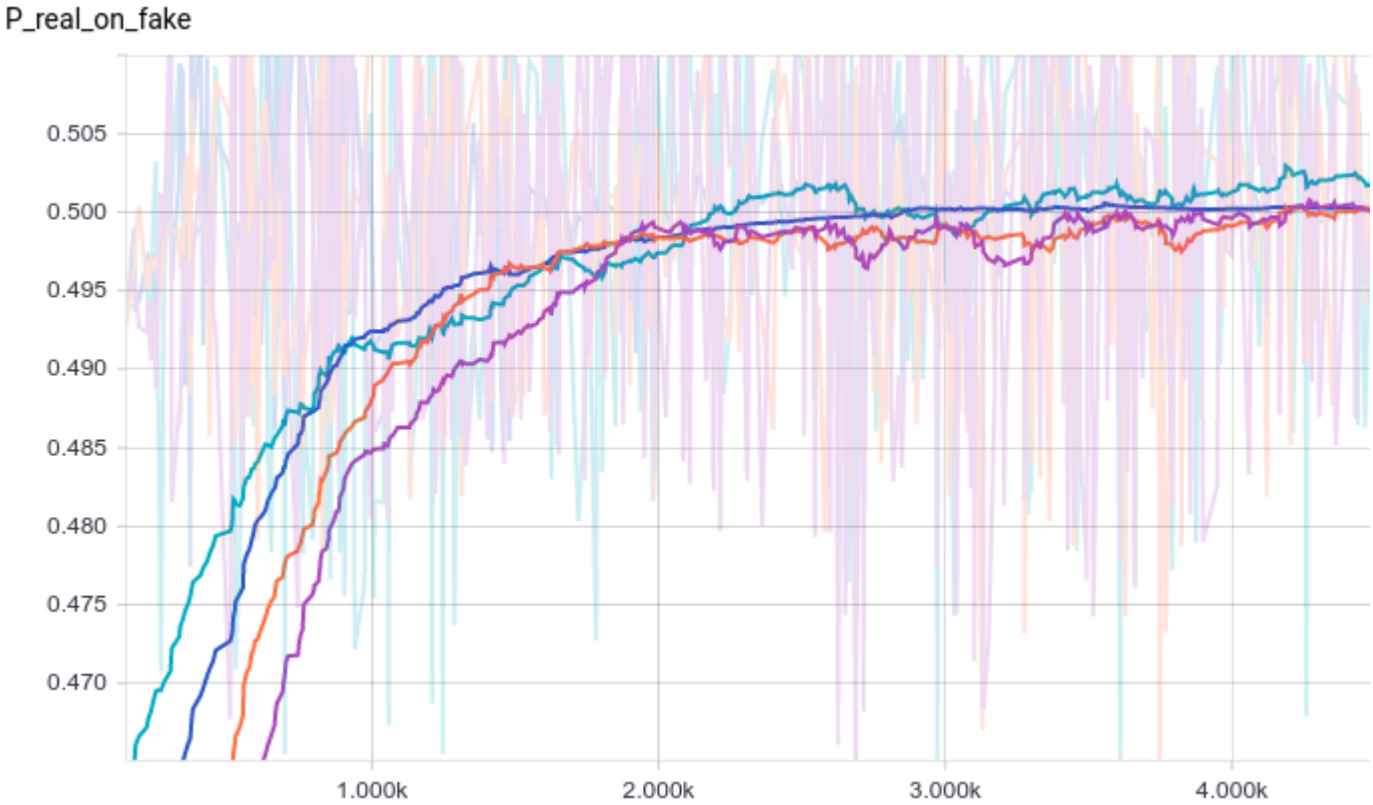

рис.5 Вероятность классификации дискриминатором реального примера как реального.

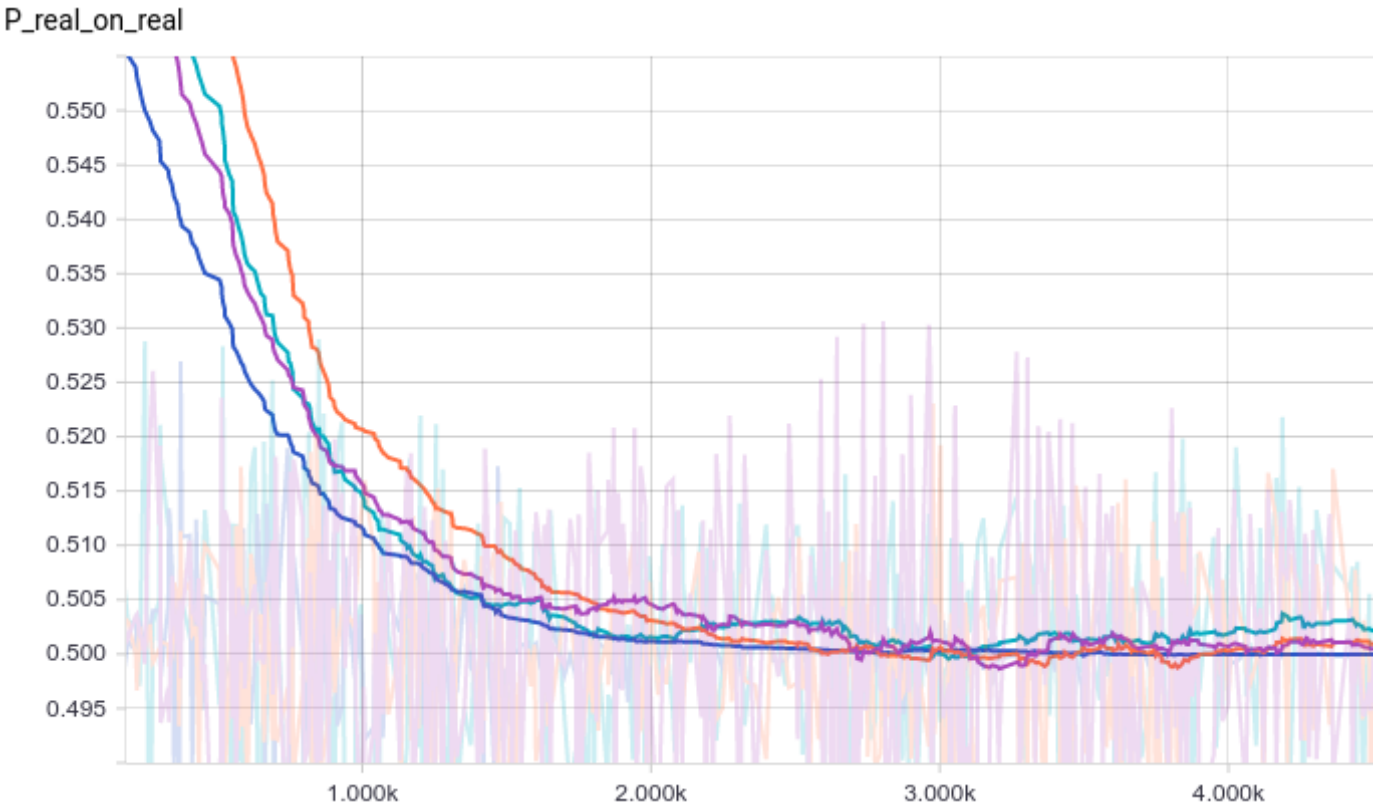

рис.6

Видно, что из за большого количества параметров обучение стало гораздо более шумным. Дискриминаторы всех моделей сходятся к результату около image, но ведут себя нестабильно вокруг этой точки. Давайте посмотрим на форму генератора. Его форма похожа на распределение *TwinPeaks*.

Самая регуляризованная модель показала себя лучше всех. Она выучила две моды, примерно совпадающие с модами распределения данных. Размеры пиков тоже не очень точно, но приближают распределение данных. Таким образом, нейросетевой генератор способен выучить мультимодальное распределение данных.

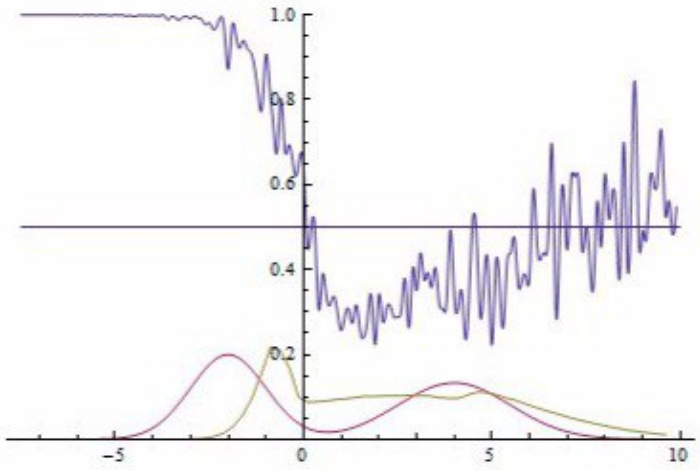

рис.7

Итак, если использовать более сложное устройтство генератора и обучаться на больших данных, то можно будет значительно улучшить показания на метриках качества.

8 Список литературы

1. Stein, B., Barrón Cedeño, L.A., Eiselt, A., Potthast, M., Rosso, P.: Overview of the 3rd international competition on plagiarism detection. In: CEUR Workshop Proceedings. CEUR Workshop Proceedings (2011)
2. http://pan.webis.de/clef18/pan18-web/author-identification.html
3. H. A. Chowdhury, D. K. Bhattacharyya, Plagiarism: Taxonomy, tools and detection techniques, arXiv. URL https://arxiv.org/pdf/1801.06323.pdf
4. https://pdfs.semanticscholar.org/1011/6d82a8438c78877a8a142be47c4ee8662138.pdf
5. https://arxiv.org/pdf/1701.06547.pdf
6. Zechner, M., Muhr, M., Kern, R., Granitzer, M.: External and intrinsic plagiarism detection using vector space models. Proc. SEPLN. vol. 32 (2009)
7. Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by authorship within and across documents. CEUR Workshop Proceedings (2016)
8. https://pdfs.semanticscholar.org/c70e/7f8fbc561520accda7eea2f9bbf254edb255.pdf
9. http://pan.webis.de/clef18/pan18-web/author-identification.html
10. http://www.mathnet.ru/links/21c7959c3887dcf64bc0f1b5913c81be/ia487.pdf
11. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles (2009)
12. https://pdfs.semanticscholar.org/c70e/7f8fbc561520accda7eea2f9bbf254edb255.pdf
13. https://cyberleninka.ru/article/v/analiz-metodov-binarnoy-klassifikatsii
14. https://cyberleninka.ru/article/v/metod-maksimalnogo-pravdopodobiya-v-prilozhenii-k-lchm-signalam