Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»
Факультет компьютерных наук

Васильев Руслан Леонидович

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Методы обнаружения текста, сгенерированного большими языковыми моделями

Text Generated by Large Language Models: Detection Techniques

по направлению подготовки 01.04.02 Прикладная математика и информатика
образовательная программа «Современные компьютерные науки»

Научный руководитель:
д.ф-м.н., доцент
А. Г. Дьяконов

Студент:
Р. Л. Васильев

Москва, 2024

Содержание

Аннотация

Стремительное развитие и распространение больших языковых моделей (LLM) делают необходимой разработку надежных и эффективных инструментов для детекции сгенерированного текста. Тексты, написанный с помощью LLM, грамматически корректны, обладают логикой и связностью, поэтому их обнаружение может быть нетривиальным. В данной работе производится сравнение известных алгоритмов детекции в различных сценариях, в том числе на out-of-domain данных. Показано, что ансамблирование повышает робастность базовых методов. Предложен и опубликован новый датасет[1], для которого были использованы state-of-the-art языковые модели: GPT4 Turbo, GPT4 Omni, Claude 3 Opus, Llama3, CommandR+, YandexGPT, GigaGhat — суммарно для датасета было сгенерировано 21 000 примеров. Представлен новый метод для обучения детекторов, использующий Pairwise Ranking Loss, который показывает более высокое качество по сравнению со стандартным fine-tuning на бинарную разметку. Исходный код экспериментов опубликован[2].

Abstract

The rapid development and widespread adoption of large language models (LLMs) necessitate the creation of reliable and effective tools for detecting machine-generated text. Texts written using the LLMs are gramatically correct, logical and coherent, making their detection non-trivial. This work conducts a comparison study of known detection algorithms in different scenarios, including out-of-domain data. The results indicate that ensembling strengthens the robustness of the methods. A new dataset has been proposed and published[1], for which state-of-the-art LLMs were used: GPT4 Turbo, GPT4 Omni, Claude 3 Opus, Llama3, CommandR+, YandexGPT, GigaGhat — in total, 21 000 texts were generated for the dataset. A new method for training detectors using Pairwise Ranking Loss is presented, which demonstrates higher quality compared to standard BCE fine-tuning. The source code of the experiments has been published[2].

[1] `https://huggingface.co/datasets/artnitolog/llm-generated-texts`
[2] `https://github.com/artnitolog/llm-detection-techniques`

1 Введение

В последние годы большие языковые модели *(large language models, LLM)* начали достигать достигать человеческого уровня во многих задачах [1, 2]. Современные LLM способны генерировать тексты, которые на первый взгляд неотличимы от написанных человеком [3] и могут в том числе превосходить их по связности и грамматической корректности. С таким стремительным прогрессом особенно остро встает вопрос о возможности обнаруживать тексты, сгенерированные большими языковыми моделями.

Языковые модели могут использоваться для генерации фейковых новостей и распространения дезинформации [4], поддержки поддельных аккаунтов в социальных сетях [5], а также создания токсичного и вредоносного контента [6]. Даже при разумном использовании LLM с распространением сгенерированного контента снижается доверие к публикуемой информации, поскольку языковые модели склонны ошибаться и придумывать факты [7]. Кроме прямых рисков для общества, сгенерированные тексты могут негативно влиять на следующие итерации обучения самих LLM [8].

Проблема обнаружения текста, сгенерированного большими языковыми моделями, связана с разработкой различных подходов и методов, направленных на умение отличать человеческие тексты от сгенерированных [9, 10, 11, 12]. Наибольшую популярность приобрели методы, основанные на подсчете различных статистик в качестве признаков, чаще всего с использованием сторонних языковых моделей [10, 13, 14, 11], а также основанные на обучении классификаторов, основанных на трансформерных архитектурах [15, 3, 16, 17].

Несмотря на прогресс в области алгоритмов детекции, общая проблема остается нерешенной и усложняется с выходом новых языковых моделей, генерации которых становятся все более качественными. В данной работе приводится систематическое исследование различных методов, предлагается новый бенчмарк качества и улучшение одной из наиболее сильных стратегий детекции. Опишем структуру дальнейшей работы:

- В разд. 2 задача обнаружения сгенерированного текста ставится формально, вводятся условия и ограничения.
- В разд. 3 описываются критерии качества, использующиеся для измерения эффективности детекторов.
- В разд. 4 обозреваются имеющиеся датасеты и описывается, как был собран построенный в рамках данной работы бенчмарк.
- В разд. 5 приводится описание известных методов детекции, а также предлагается новый алгоритм, улучшающий стандартный fine-tuning.
- В разд. 6 описывается дизайн и результаты поставленных экспериментов, как in-domain, так и out-of-domain.

2 Постановка задачи

Поставим формально задачу обнаружения (детекции) текста, сгенерированного большими языковыми моделями [12].

Под *текстом, сгенерированным большой языковой моделью (LLM-generated text)* понимается результат работы текстовой генеративной нейросети, представляющий собой связный, чаще всего грамматически корректный текст. Обычно такой текст является ответом на некоторый запрос *(prompt, request)* или историю запросов пользователя.

Текст, написанным человеком (human-written text) выражает уже человеческие мысли: примерами являются художественная литература, научные статьи, новости, переписки, документы, и т.д.

Задача *обнаружения (детекции) текста, сгенерированного большими языковыми моделями (LLM-generated text detection)* [12] формулируется как задача бинарной классификации. От детектора $\mathcal{D}$ ожидается ответ на вопрос, был ли текст x сгенерирован большой языковой моделью:

$$\mathcal{D}(x) = \begin{cases} 1, & \text{если x сгенерирован LLM;} \\ 0, & \text{если x написан человеком.} \end{cases} \tag{1}$$

Существует 2 основных направления детекции: *white box* и *black box*. В первом случае дополнительно предполагается, что кроме текста x известны его логиты *(логарифмы вероятностей токенов)*, посчитанные целевой LLM, в последнем же — только текст. Для современных высококачественных проприетарных LLM, результаты которых особенно важно уметь обнаруживать, white-box сценарий оказывается нерелевантным, поскольку такие LLM предоставляют API исключительно на генерацию. Поэтому далее в работе рассматривается именно black-box сценарий.

Кроме того, в задаче детекции оказывается важным размер текстов: на коротких обнаружить след генеративной модели сложнее, чем на длинных [3]. Далее в экспериментах мы рассмотрим обе постановки, сравнив методы как на длинных, так и на коротких текстах.

3 Критерии качества

Поскольку (1) представляет собой стандартную задачу классификации, качество детекторов можно оценивать стандартными функционалам качества бинарной классификации.

Один из наиболее легко интерпретируемых показателей качества — accuracy — доля верно классифицированных текстов:

$$\text{Accuracy} = \frac{TP}{TP + FP + TN + FN}, \tag{2}$$

где:

- TP — количество текстов, которые были сгенерированы LLM и корректно классифицированы детектором;
- FP — количество текстов, которые были написаны человеком, но неверно определены детектором как сгенерированные;
- TN — количество текстов, которые были написаны человеком и корректно классифицированы детектором;

- TP — количество текстов, которые были сгенерированы LLM, но неверно определены детектором как человеческие.

Accuracy часто измеряется при сравнении проприетарных детекторов [18, 19], что может быть связано именно с простотой функционала.

Довольно удобно для замера качества детекции использовать AUC ROC — площадь под кривой ошибок или доля пар текстов вида *(человеческий, сгенерированный)*, которые алгоритм верно упорядочил. К преимуществу функционала можно отнести отсутствие зависимости от порога классификации. AUC ROC использовался, например, в соревновании, посвященному детекции текстов, сгенерированных LLM [20].

Однако чаще всего в бенчмарках [16, 21, 3, 22] замеряется F1-мера, являющаяся среднегармоническим точности (Precision) и полноты (Recall):

$$\text{F1} = 2 \cdot \frac{1}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}, \tag{3}$$

где:

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}. \tag{4}$$

Точность в задаче (1) позволяет оценить, какая доля текстов, определенных алгоритмом как сгенерированные LLM, действительно были сгенерированы. Высокая точность, то есть низкая доля FP, важна, в частности, в задаче детекции студенческих эссе [23].

Recall (4) же позволяет оценить, сколько алгоритму удалось обнаружить текстов, сгенерированных LLM, среди всех сгенерированных текстов. Преимущество F1-меры (3) заключается в том, что он сильно чувствителен к понижению точности или полноты по отдельности. В дальнейшем в замерах качества методов детекции мы будем использовать именно F1-мера.

4 Датасеты и бенчмарки

4.1 Имеющиеся работы

Датасеты и бенчмарки качества необходимы для честного сравнения алгоритмов детекции. Для построения таких датасетов обычно используют фиксированные запросы, которым соответствуют пары текстов: написанные человеком и сгенерированные языковыми моделями. Таким образом, датасеты являются параллельными.

Общие датасеты с разными доменами и/или моделями стали появляться вместе с распространением GPT-подобных моделей. Одним из известных примеров является TuringBench [21]: в нем использовано множество моделей, вышедших в 2018–2020 — с тех пор качество LLM значимо выросло, поэтому сейчас датасет практически не встречается в замерах.

С выходом ChatGPT[3] внимание к проблеме обнаружения сгенерированных текстов усилилось: было предложено несколько бенчмарков с современными LLM. Одним из наиболее известных стал датасет HC3 [3] с десятками тысяч примеров,

[3] `https://chat.openai.com/chat`

сгенерированных ChatGPT и написанных людьми. У него есть и улучшенная версия: HC3+ [24] с генерациями GPT-3.5-Turbo. Однако в данных датасетах ограничено разнообразие: тексты сгенерированы единственной LLM.

Более разнообразные бенчмарки были предложены в последующих работах. В [22] и [25] собраны данные на различных языках и LLM. Популярным бенчмарков сейчас является MGTBench [16]: в нем содержатся примеры, сгенерированные ChatGLM, Dolly, ChatGPT, GPT4All, StableLM и Claude на трех различных доменах. Этот бенчмарк в свою очередь является расширением датасета [13]. Бенчмарк MGTBench разнообразен, но имеет 2 особенности:

- На текущий момент использованные в нем LLM далеки от state-of-the-art: конце 2023 – начале 2024 вышло много более качественных моделей *(конкретные модели рассмотрим далее)*;
- Часть примеров бенчмарка не является *параллельной*: многие человеческие тексты не соответствуют запросам, использовавшимся для генерации с помощью LLM.

В предложенном датасете мы решаем данные проблемы, собирая параллельный бенчмарк из текстов, сгенерированных 7-ю современными LLM.

4.2 Предложенный датасет

Для генерации использованного в работе датасета мы используем методолгию и запросы [13, 16], но используем state-of-the-art LLM и восстанавливаем параллельность, которая потребуется в предложенном для детекции методе — Pairwise Ranking Fine-Tuning. В разд. 4.2.1 описаны источники данных: запросов *(prompts)* и человеческой разметки, в разд. 4.2.2 —– большие языковые модели, использованные для генерации датасета. Примеры запросов, человеческих текстов и генераций представлены в табл. 1.

4.2.1 Источники

Student Essays Для данного источника использовался корпус эссе IvyPanda, написанный школьниками и студентами, датасет отличается большим разнообразием тем и дисциплин [4]. У датасета отсутствует разметка в виде заголовков или вопросов, которые можно было бы использовать в качестве запроса, поэтому изначально авторы [13] отдельно сгенерировали соответствующие темы с помощью ChatGPT. Далее для каждого такого запроса генерировалась эссе с помощью ChatGPT и других LLM. Однако впоследствии в качестве человеческих текстов были взяты другие эссе, не соответствующие запросам. Для поиска исходных студенческих эссе в данной работе была использована модель для семантического поиска[5]: в качестве запросов *(queries)* были использованы как раз сгенерированные темы *(prompts)*, в качестве индекса — исходный датасет IvyPanda.

[4] https://huggingface.co/datasets/qwedsacf/ivypanda-essays
[5] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Источник	Запрос	Человеческий текст	GPT4 Omni	YandexGPT 3 Pro
Student Essays	Write an essay in 700 words to the prompt "To what extent did television shape the rock and pop music scene in the 1960s?."	The rock and music pop scene were introduced to and gained popularity on television through massive hit shows for teenagers. The three primary ones were Where the Action Is on CBS, Shinding on ABC, an...	The 1960s were a transformative decade marked by profound social, political, and cultural changes. Within this vibrant landscape, the rock and pop music scene burgeoned, heavily influenced by a variet...	**The Impact of Television on Rock and Pop Music in the 1960s** The 1960s were a time of great cultural change, and television played a significant role in shaping the rock and pop music scene of tha...
News Articles	Write a news article in 600 words based on the headline "North American metals producers to face weakened third-quarter earnings, says Reuters survey."	Weak metals prices will undermine third-quarter earnings for many North American metals producers, analysts surveyed by Reuters said. "In most cases, we'll see they're going to be down, either year ov...	**Headline: North American Metals Producers to Face Weakened Third-Quarter Earnings, Says Reuters Survey** In a period marked by mounting economic uncertainty and fluctuating commodity prices, North ...	**North American metals producers to face weakened third-quarter earnings, says Reuters survey** The Reuters survey has revealed that North American metals producers are likely to face weakened earni...
Creative Writing	Write a story in 400 words to the prompt "You are born with the ability to stop time, but one day you see something else is moving when you have already stopped time.."	If you could have one power, what would it be? Time control, that was my preference. The ability to stop anyone, anything, and alter what is to happen. My personal favourite, was freezing time while i...	Ever since I was a child, I possessed the inexplicable ability to stop time. With a mere thought, the world around me would come to a halt. My mother's laughter would freeze mid-giggle, rain suspended...	I was born with the ability to stop time. It was a gift that I had always taken for granted. I could stop time whenever I wanted, and it was like being in a dream. Everything would freeze, and I could...

Таблица 1: Примеры текстов из собранного датасета

News Articles Здесь в качестве исходного корпуса использовался [26], изначально собранный для задачи идентификации авторов. Датасет представляет собой разнообразные новостные статьи, написанные разными авторами. Запросы для данного датасеты были сгенерированы в [13], а в данной работе были восстановлены их исходные человеческие тексты с помощью семантического поиска аналогично «Student Essays».

Creative Writing Заключительный датасет основан на форуме WritingPrompts[6], в котором различные авторы придумывают истории на заданные темы. В данном источнике запросы и человеческая разметка изначально параллельные.

Отметим, что для каждого текста имеется соответствующий ему запрос [13], в котором в частности указано примерное (округленное до 100) количество слов в человеческом тексте. Такой подход использовался для того, чтобы минимизировать влияние длины текста на детекторы.

4.2.2 Модели

Далее для генерации текстов нами были использованы LLM, вышедшие в конце 2023 — начале 2024. GPT4 Turbo[7], GPT4 Omni[8] и Claude 3 Opus[9] выбраны как прориетарные модели с открытым API, имеющие наиболее высокое качество точки зрения лидерборда LLM [27]. Llama3 70B[10] и Command R+[11] выбраны по тому же критерию, но среди моделей с открытым исходным кодом и опубликованными весами. Кроме того, в бенчмарке представлены YandexGPT 3 Pro[12] и GigaChat Pro[13] — известные российские LLM.

Из Student Essays, News Articles и Creative Writing было взято по 1 000 запросов, каждый из которых использовался для генерации текста с помощью рассмотренных моделей на трех доменах. Таким образом, всего в рамках предложенного бенчмарка сгенерированы 21 000 текстов.

5 Методы

5.1 Существующие подходы

5.1.1 Probability

Языковые модели генерируют текст авторегрессионно, оценивая вероятностное распределение каждого следующего i-го токена x_i на основе контекста:

$$p_\theta(x_i \mid x_{<i}) = p_\theta(x_i \mid x_1, \ldots, x_{i-1}). \tag{5}$$

[6] https://www.reddit.com/r/WritingPrompts/

[7] GPT-4 Turbo 2024-04-09, OpenAI: https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4

[8] GPT-4 Omni, OpenAI: https://openai.com/index/hello-gpt-4o

[9] Claude 3 Opus, Anthropic: https://www.anthropic.com/news/claude-3-family

[10] Llama3 70B, Meta: https://llama.meta.com/llama3

[11] Command R+, Cohere: https://cohere.com/blog/command-r-plus-microsoft-azure

[12] YandexGPT 3 Pro, Yandex: https://ya.ru/ai/gpt-3

[13] GigaChat Pro, SberDevices: https://developers.sber.ru/portal/news/giga_chat_pro-15-12-2023

Вероятность полного текста x с точки зрения модели θ выражается следующим образом:

$$p_\theta(x) = \prod_i p_\theta(x_i \mid x_{<i}). \tag{6}$$

Поскольку на обучении LLM фактически оптимизируют (6), а во время генерации [28] на каждом шаге выбирают токены, соответствующие высоким значениям (top-k, top-p (5)), у сгенерированных текстов оценки вероятности оказываются выше [10].

На практике оценка (6) дополнительно логарифмируется. Метод проиллюстрирован на рис. 1.

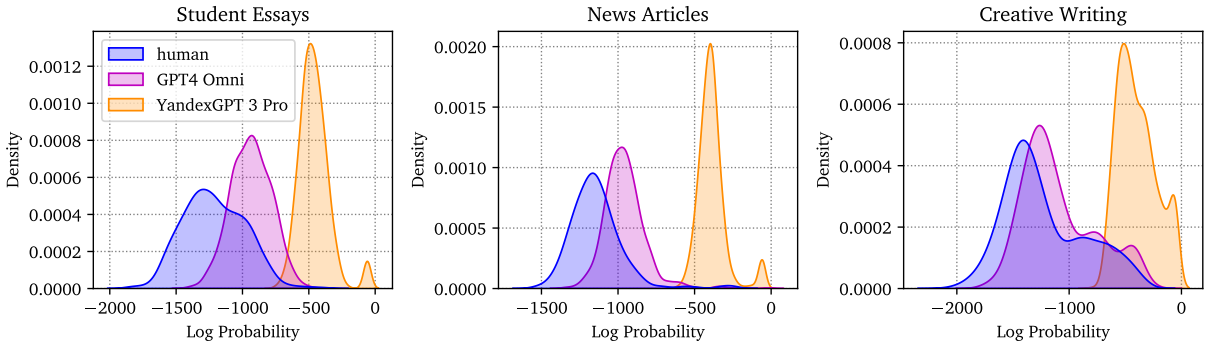

Рис. 1: Распределение оценок, полученных методом разд. 5.1.1

В качестве оценок для θ зачастую берется другая LLM, суррогатная *(surrogate)*, поскольку вероятности исходной модели недоступны. Всюду в данной работе в качестве суррогатной языковой модели для иллюстраций и замеров качества использована Gemma 2B [29].

5.1.2 Perplexity

Вместо оценки вероятности текста (6) в качестве бейзлайна чаще [3, 16, 13] используется логарифм перплексии:

$$\text{Perplexity} = \frac{1}{\sqrt[n]{\prod_{i=1}^{n} p_\theta\left(x_i \mid x_{<i}\right)}}. \tag{7}$$

Здесь n — длина последовательности в токенах. На практике выражение (7) также логарифмируется. Перплексия используется как базовый функционал качества языковых моделей и формально ее логарифм соответствует кросс-энтропии между протокенизированным текстом и распределением на выходе модели.

Качественное отличие от разд. 5.1.1: в перпрексии производится нормировка на длину текста. Распределение оценок метода проиллюстрировано на рис. 2.

5.1.3 Rank

В [14] было предложено использовать усредненные ранги токенов. Под рангом токена x_i понимается его порядок $r_\theta(x_i)$ при сортировке вероятностей по словарю. Аналогично предыдущим методом, основная интуиция метода связана с тем, что

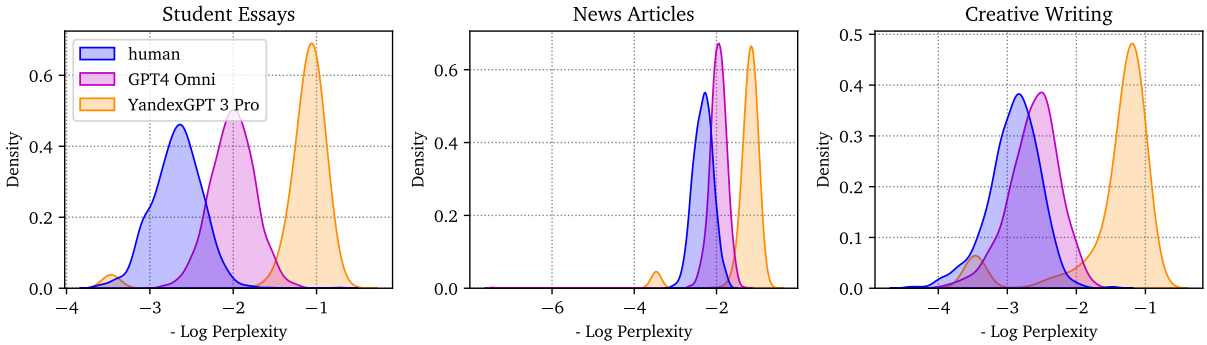

Рис. 2: Распределение оценок, полученных методом разд. 5.1.2

токены сгенерированных текстов будут более вероятны с точки зрения LLM, а значит чаще будут оказываться среди начальных токенов при сортировке по словарю рис. 3.

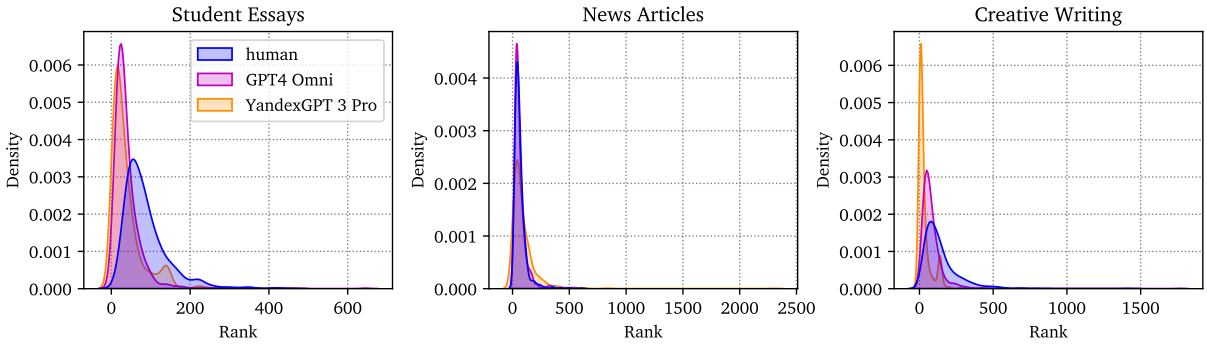

Рис. 3: Распределение оценок, полученных методом разд. 5.1.3

5.1.4 LogRank

В [11] было предложено усреднять не ранги токенов, а их логарифмы, поскольку относительный порядок маловероятных токенов не сильно информативен (рис. 4).

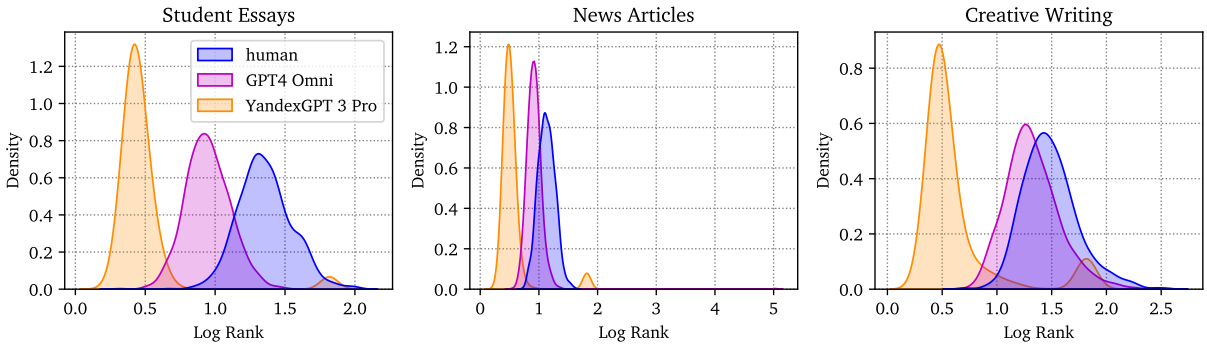

Рис. 4: Распределение оценок, полученных методом разд. 5.1.4

5.1.5 Entropy

Авторы [14] обнаружили, что также в качестве метода детекции можно находить оптимальный порог по энтропии распределения (рис. 5). Энтропия считается для

предсказания на каждый следующий токен, а далее усредняется по всей последовательности:

$$\text{Entropy(x)} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{V} p_\theta(\tilde{x}_j \mid x_{<i}) \log p_\theta(\tilde{x}_j \mid x_{<i}), \tag{8}$$

где V — размер словаря, $\tilde{x}_j$ — j-й токен в словаре, $p(\tilde{x}_j \mid x_{<i})$ — оценка вероятности токена $\tilde{x}_j$ при условии контекста $(x_1, \ldots, x_{i-1})$ *(префикс исходного текста)*.

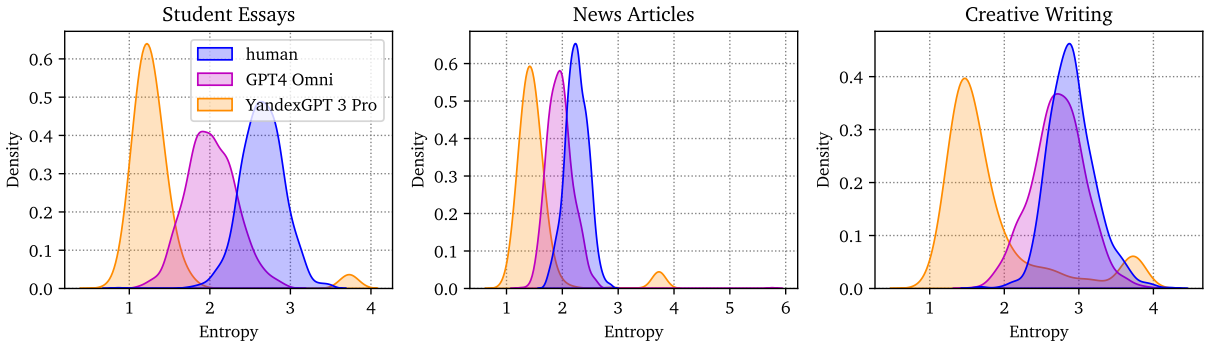

Рис. 5: Распределение оценок, полученных методом разд. 5.1.5

5.1.6 GLTR

GLTR [14] совмещает в себе несколько статистик, формируя четырехмерные векторы в качестве признаков. А именно, считаются доли токенов, попавших соответственно в бины: top10, top100, top1000 и все оставшиеся. Такой набор признаков оказывается достаточно информативным: у сгенерированных текстов токены чаще попадают именно в первые бины. Признаки далее используются для обучения логистической регрессии.

5.1.7 DetectGPT

В [11] было замечено, что сгенерированные тексты часто лежат в локальных оптимумах правдоподобия с точки зрения языковых моделей. А именно, небольшие возмущения *(perturbations)* по-разному влияют на сгенерированные и человеческие тексты: правдоподобие первых при возмущениях падает, в то время как у человеческих такой тенденции нет. Для построение детектора авторы замеряют следующую величину:

$$\mathbf{d}(x, p_\theta, q) = \log p_\theta(x) - \mathbb{E}_{\tilde{x}\sim q(\cdot|x)} \log p_\theta(\tilde{x}). \tag{9}$$

В выражении (9) *(discrepancy)* $\log p_\theta(x)$ соответствует логарифму правдоподобия исходного текста, а $\tilde{x}$ — тексту, возмущенному с помощью некоторой модели q *(обычно модели T5 [30])*. Таких возмущений для точной оценки оценки требуется достаточно много: в экспериментах авторы используют 100 возмущенных текстов, по которым впоследствии оценивается $\mathbb{E}_{\tilde{x}\sim q(\cdot|x)} \log p_\theta(\tilde{x})$.

Именно генерация возмущений q делает метод (9) вычислительно дорогим: каждое возмущение соответствует вызову другой языковой модели, поэтому использование DetectGPT в практических применениях может быть затруднительным.

5.1.8 DetectLLM

В [31] было фактически объединяются методы разд. 5.1.3 и разд. 5.1.7. В качестве критерия, аналогичного (9) предлагается NPR *(normalized log-rank perturbation)*:

$$\text{NPR} = \frac{\mathbb{E}_{\tilde{x}\sim q(\cdot|x)} \log r_\theta(\tilde{x})}{\log r_\theta(x)}, \tag{10}$$

где $r_\theta(x)$ — средний ранг токенов исходной последовательности x, а $\mathbb{E}_{\tilde{x}\sim q(\cdot|x)} \log r_\theta(\tilde{x})$ — усредненные ранги по возмущения q. Как и (9), оценка (10) вычислительно дорогая из-за необходимости вызывать большое число генераций $\tilde{x}$.

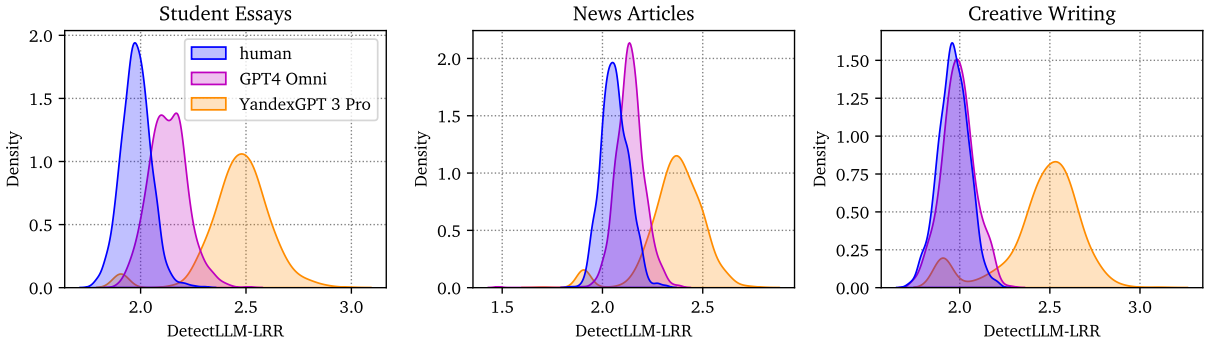

Рис. 6: Распределение оценок, полученных методом разд. 5.1.8 (LRR)

Как альтернативу, авторы предложили более легковесный метод LLR *log-likelihood log-rank ratio*:

$$\text{LRR} = \frac{\sum_{i=1}^{n} \log p_\theta(x_i \mid x_{<i})}{\sum_{i=1}^{n} \log r_\theta(x_i \mid x_{<i})}, \tag{11}$$

которое фактически представляет собой частное оценок разд. 5.1.2 и разд. 5.1.4. Метод проиллюстрирован на рис. 6.

5.1.9 FastDetectGPT

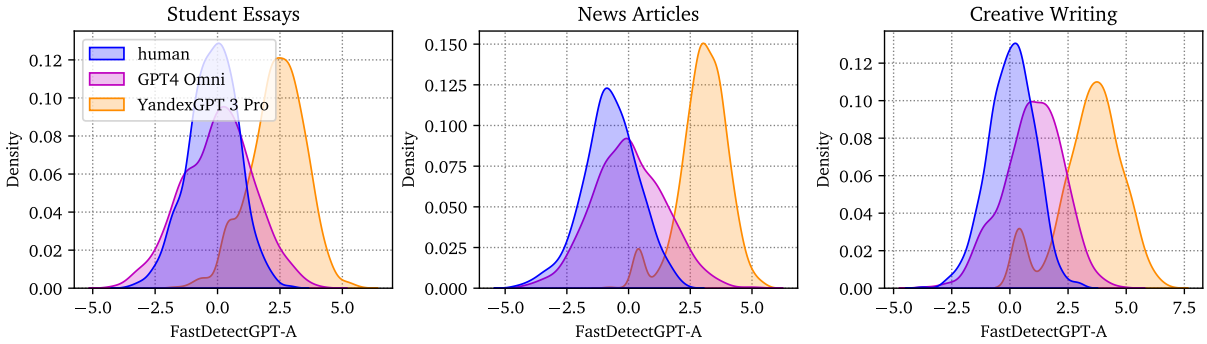

Рис. 7: Распределение оценок, полученных методом разд. 5.1.9 (LRR)

В методе [32] развивается предположение о том, что сгенерированные тексты чаще оказываются в локальном оптимуме, но по сравнению с (9) для оценки возмущений не используются многократные вызовы генеративной модели. Авторы предлагают следующую оценку:

$$\mathbf{d}(x, p_\theta, q_\varphi) = \frac{\log p_\theta(x|x) - \tilde{\mu}}{\tilde{\sigma}}, \tag{12}$$

где

$$\tilde{\mu} = \mathbb{E}_{\tilde{x}\sim q_\varphi(\tilde{x}|x)}\left[\log p_\theta(\tilde{x}|x)\right], \qquad \tilde{\sigma}^2 = \mathbb{E}_{\tilde{x}\sim q_\varphi(\tilde{x}|x)}\left[(\log p_\theta(\tilde{x}|x) - \tilde{\mu})^2\right]. \tag{13}$$

В выр. 12 $\log p_\theta(x|x)$ соответствует логарифму правдоподобия исходного текста. В качестве распределения $q_\varphi(\cdot|x)$ берутся выходные вероятности при подаче текста x в суррогатную модель: то есть фактически мы обуславливаемся на весь исходный контекст, но можем зашумить каждый токен независимо. Такой подход позволяет единожды вызвать модель θ в отличие многократных генераций в (9).

Матожидания в (13) соответствуют усреднению по 10 000 семплированиям *(FastDetectGPT-S)*, однако авторы [32] показывают, что можно использовать и аналитическую оценку *(FastDetectGPT-A)*, а сам метод не только значительно быстрее разд. 5.1.7, но и работает качественнее. Оценки (12) проиллюстрированы на рис. 7.

5.1.10 BCE Fine-Tuning

Языковые модели, основанные на транфсормерной архитектуре, при дообучении на размеченных данных *(fine-tuning)* достигают высокого качества в задачах классификации текста [33, 34]. В нашем случае мы имеем классификацию на 2 класса: сгенерированный текст или написанный человеком, выр. 1. В случае бинарной классификации стандартное дообучение соответствует минимизации следующей функции потерь:

$$\mathcal{L}_{\text{BCE}} = \frac{1}{N}\sum_{i=1}^{N}\left[-y_i \log \widehat{p}\,(x_i) - (1 - y_i)\log\left(1 - \widehat{p}\,(x_i)\right)\right], \tag{14}$$

где N — число текстов x_i в батче, y_i — метка класса в соответствии с (1). $\widehat{p}\,(x_i)$ моделируется с помощью добавления линейного слоя к трансформеру, который отображает последнее скрытое представление в скаляр, и сигмоиды $\sigma(s) = \frac{1}{1+e^{-s}}$.

Модели, обученные на (14), нередко оказывались наиболее качественными детекторами в прежних работах [15, 3, 16, 17], но при этом по сравнению с более простыми моделями их обобщающая способность может хуже переноситься на тексты других доменов.

5.2 Предложенные подходы

Рассмотренные ранее методы делятся на 2 группы: метрические (*metric-based*: разд. 5.1.1 – разд. 5.1.9) и модельные (model-based: разд. 5.1.10). С точки зрения обучения классификатора первые сильно дешевле, поскольку требуют только применения модели, последние же требуют ресурсов на полноценное дообучение (fine-tuning). Ввиду множества сценариев детекции оба направления остаются актуальными.

5.2.1 Metric Ensemble

Большая часть методов детекции (разд. 5.1.1 – разд. 5.1.6, разд. 5.1.9) представляет некоторое преобразование над логитами суррогатной модели. В рамках

Metric Ensemble мы предлагаем объединить все рассмотренные величины в единое признаковое пространство и обучить на них легковесную модель *(например, логистическую регрессию)*.

Отметим, что вычислительно такой ансамбль практически не замедляет исходные методы детекции, поскольку самая дорогая операция, вычисление логитов, выполняется единожды.

5.2.2 Pairwise Ranking Fine-Tuning

Заметим, что датасеты (разд. 4) для обучения детекторов и замера качества чаще всего параллельные: отдельному запросу соответствует как текст x_p, сгенерированный LLM, так и текст x_n, написанный человеком. На таких парах мы ожидаем, что детектор будет оценивать $\widehat{p}\left(x_p\right) > \widehat{p}\left(x_n\right)$, и предлагаем учиться именно на параллельных парах, используя следующую функцию потерь для отдельной пары:

$$\mathcal{L}_{\text{PR}} = -\log\left(\sigma\left(\log\widehat{p}\left(x_p\right) - \log\widehat{p}\left(x_n\right)\right)\right). \tag{15}$$

Ранжирующая функция потерь (15) позволяет учиться разделять семантически близкие тексты, в отличие от (14), в которой решается более сложная задача разделения произвольных текстов. Чтобы учесть и глобальные, и локальные отличия в рамках итогового алгоритма предлагается комбинированная функция потерь:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \alpha\mathcal{L}_{\text{PR}}, \tag{16}$$

где α — гиперпараметр.

Функционал (15) мотивирован не только стратегией сбора данных. В [3] было показано, что люди с более высоким качеством способны разделять именно пары текстов, соответствующих одному запросу *(в рамках статьи — некоторому вопросу)*, и хуже классифицируют произвольные тексты.

Кроме того, функционал (15) в последние годы стал стандартным для моделей награды в обучении с подкреплением [35, 36] при оптимизации LLM на следование человеческим предпочтениям. Отметим, что дополнительно можно было бы обуславливаться на запрос *(prompt)*, но в рамках данной работы все методы детекции принимают на вход только текст, сгенерированный LLM или написанный человеком.

6 Результаты

В экспериментах методы учатся на отдельных источниках из разд. 4.2.1 и LLM из разд. 4.2.2, далее замеряются как in-domain, то есть на тех же источниках и LLM (разд. 6.1), так и на out-of-domain. В качестве out-of-domain мы используем несколько постановок:

- Cross-Datasource (разд. 6.2) — замена источника даннах *(например, обучение на студенческих эссе и замер на новостном датасете)*;
- Cross-LLM (разд. 6.3) — замена детектируемой LLM *(например, обучение детектора на GPT4 и замер на YandexGPT)*.

Отдельно рассматривается детекция на коротких текстах *(в данном сценарии тексты из обучающей и тестовой выборки обрезаются до 50 слов)*, как более сложная задача, и на полных текстах, как более простая задача.

Для обучения детекторов *(для подбора порогов, обучения логистической регрессии, а также обучения транфсормера)* используется 80% датасета, оставшиеся 20% составляют тестовую выборку. Все представленные функционалы качества соответствуют усреднению по 3-м случайным разбиениям датасета для снижения шума.

В алгоритмах, основаных на логитах суррогатной модели, используется языковая модель Gemma 2B [29]. В *Metric Ensemble* в качестве модели используется логистическая регрессия. В случае дообучения (fine-tuning) мы используем гиперпараметры [16], в качестве базовой модели взята модель [37]. В (16) используется значение $\alpha = 0.1$, сработавшее лучше на сетке $\{0.0, 0.1, 0.5, 1.0\}$ в предварительных экспериментах.

Код обучения и замеров со всеми гиперпараметрами опубликован, таким образом, эксперименты могут быть воспроизведены[2].

6.1 In-Domain

Результаты бенчмарков приведены в табл. 2–табл. 7 (отметим, что в таблицах использованы сокращения в названиях методов: Proba — Probability разд. 5.1.1, PPL — Perplexity разд. 5.1.2, LRank — LogRank разд. 5.1.4, FD — FastDetectGPT разд. 5.1.9, ME — Metric Ensemble, BCE — BCE Fine-Tuning разд. 5.1.10, PR — Pairwise Ranking Fine-Tuning разд. 5.2.2).

Как видим, в in-domain сценарии у методов, основанных на fine-tuning качество выше других методов, причем практически на всех моделях и источниках у Pairwise Ranking F1 выше стандартного BCE.

Отметим, что среди известных методов без fine-tuning не выделяется однозначно лучший критерий. Например, на срезе студенческих эссе (табл. 2, табл. 5) лучше всех сработало разделение по перплексии. В коротких новостных статьях и датасете с историями (табл. 3, табл. 4) — FastDetectGPT. В случае длинных текстов в новостях лучше сработала перплексия, в «Creative Writing» — GLTR.

Однако предложенный Metric Ensemble, объединяющий все рассмотренные критерии, достигает максимального качества среди metric-based методов на всех рассмотренных постановках, а в части моделей оказывается даже лучше fine-tuning.

Наименее оптимальные критерии также зависят от датасета. В коротких студенческих эссе (табл. 2) хуже всего сработал Rank-критерий, на длинных (табл. 5) он идет вторым с конца — у FastDetectGPT наиболее низкое качество. В случае новостей слабым критерием также оказались ранги (табл. 3, табл. 6), в том числе связанный критерий — DetectLLM-LRR. На «Creative Writing» в коротких текстах низкое качество у энтропийного критерия и Rank (табл. 4), на длинных текстах хуже сработал FastDetectGPT (табл. 7).

	GPT-4 Turbo	GPT-4 Omni	Claude3 Opus	Llama3 70B	Cmd R+	YaGPT3 Pro	Giga-Chat	Avg
Proba	67.53	68.27	88.52	89.41	80.98	89.29	84.51	81.22
PPL	70.62	71.01	87.58	89.10	81.81	88.87	82.72	81.67
Rank	59.68	56.37	65.22	65.30	64.23	44.41	61.07	59.47
LRank	67.53	67.30	86.93	88.56	81.12	87.76	82.17	80.20
Entropy	63.45	64.68	82.78	80.23	73.06	77.95	77.63	74.25
GLTR	66.05	68.18	85.28	85.35	77.89	88.00	80.53	78.75
LRR	57.47	54.78	78.16	78.73	69.75	71.88	74.22	69.28
FD-S	63.29	65.56	74.38	79.39	74.04	83.98	69.17	72.83
FD-A	63.39	65.30	74.38	79.53	74.12	84.07	69.38	72.88
ME(ours)	69.01	73.44	89.26	90.42	84.06	93.26	85.66	83.59
BCE	95.85	95.07	94.02	94.05	93.32	94.77	93.56	94.38
PR(ours)	**96.27**	**95.48**	**94.85**	**95.14**	**94.21**	**96.12**	**94.20**	**95.18**

Таблица 2: F1 методов детекции на «Student Essays», короткие тексты (до 50 слов)

	GPT-4 Turbo	GPT-4 Omni	Claude3 Opus	Llama3 70B	Cmd R+	YaGPT3 Pro	Giga-Chat	Avg
Proba	60.83	57.59	81.87	73.08	59.56	73.68	62.43	67.01
PPL	49.53	58.50	80.22	72.88	60.60	74.16	60.14	65.15
Rank	66.64	62.71	64.86	52.92	73.09	65.89	49.93	62.29
LRank	56.14	55.35	80.03	70.19	56.17	71.89	56.46	63.75
Entropy	61.51	55.23	50.56	55.76	56.00	56.32	58.64	56.29
GLTR	63.63	59.01	74.96	73.28	67.73	78.73	57.99	67.90
LRR	69.19	60.88	70.01	52.93	61.14	54.31	59.11	61.08
FD-S	62.09	69.84	88.96	89.53	72.59	86.34	73.24	77.51
FD-A	61.98	69.63	88.86	89.55	72.75	86.15	72.99	77.41
ME(ours)	84.64	88.83	90.11	93.32	87.63	94.67	79.98	88.46
BCE	98.30	**99.58**	98.10	97.99	97.18	**99.75**	**99.59**	98.64
PR(ours)	**99.08**	**99.58**	**98.35**	**98.58**	**97.28**	**99.75**	99.26	**98.84**

Таблица 3: F1 методов детекции на «News Articles», короткие тексты (до 50 слов)

	GPT-4 Turbo	GPT-4 Omni	Claude3 Opus	Llama3 70B	Cmd R+	YaGPT3 Pro	Giga-Chat	Avg
Proba	52.92	60.80	85.88	85.82	83.61	92.59	88.47	78.58
PPL	50.89	59.06	83.67	86.46	81.65	87.77	66.08	73.65
Rank	57.62	61.83	73.68	71.96	73.06	71.28	40.14	64.23
LRank	54.49	57.97	82.33	83.83	80.72	87.60	65.21	73.16
Entropy	57.55	49.69	65.19	64.92	66.12	63.71	64.21	61.63
GLTR	57.16	58.78	79.21	81.89	78.83	87.71	79.09	74.67
LRR	63.24	53.22	72.55	76.94	73.09	85.01	64.72	69.83
FD-S	53.19	64.12	86.55	90.55	81.11	91.01	66.54	76.15
FD-A	53.23	64.11	86.69	90.71	80.98	91.01	66.60	76.19
ME(ours)	68.29	72.28	89.72	92.69	87.18	**97.19**	90.89	85.46
BCE	94.73	94.36	95.49	92.66	**90.87**	95.15	94.61	93.98
PR(ours)	**94.98**	**94.54**	**96.34**	**93.45**	**90.87**	94.74	**95.35**	**94.32**

Таблица 4: F1 методов детекции на «Creative Writing», короткие тексты (до 50 слов)

	GPT-4 Turbo	GPT-4 Omni	Claude3 Opus	Llama3 70B	Cmd R+	YaGPT3 Pro	Giga-Chat	Avg
Proba	64.88	75.24	95.41	98.19	82.57	98.84	96.69	87.40
PPL	81.87	89.94	98.75	99.17	95.19	97.80	97.23	94.28
Rank	75.18	75.76	89.12	88.98	70.42	74.01	79.12	78.94
LRank	82.77	89.19	98.83	99.08	94.89	97.80	96.73	94.19
Entropy	79.28	89.07	98.59	98.76	95.09	97.48	96.39	93.52
GLTR	80.14	86.72	98.59	98.84	94.08	97.56	96.73	93.24
LRR	79.79	83.75	97.58	97.75	92.22	97.30	96.04	92.06
FD-S	52.99	52.92	66.62	86.20	62.04	88.40	66.90	68.01
FD-A	52.45	52.96	66.57	86.28	62.32	88.32	66.68	67.94
ME(ours)	84.45	90.95	98.51	99.33	95.69	**99.50**	99.09	95.36
BCE	**99.58**	99.42	**99.50**	**99.59**	99.25	98.93	**99.34**	99.37
PR(ours)	99.42	**99.83**	99.26	99.50	**99.50**	98.93	**99.34**	**99.40**

Таблица 5: F1 методов детекции на «Student Essays», длинные тексты

	GPT-4 Turbo	GPT-4 Omni	Claude3 Opus	Llama3 70B	Cmd R+	YaGPT3 Pro	Giga-Chat	Avg
Proba	54.82	78.37	97.87	98.27	81.00	99.01	98.11	86.78
PPL	59.69	80.93	99.17	99.17	84.89	97.89	94.50	88.03
Rank	45.28	59.55	84.33	70.92	58.89	45.83	21.22	55.15
LRank	58.59	79.96	99.25	99.33	83.86	97.81	94.35	87.59
Entropy	53.44	77.17	98.10	97.08	76.26	95.47	91.63	84.16
GLTR	57.46	75.37	98.18	98.59	82.20	97.40	94.75	86.28
LRR	54.68	71.26	98.24	98.41	78.54	96.02	89.49	83.81
FD-S	60.02	60.67	87.86	97.58	74.47	95.63	81.47	79.67
FD-A	60.21	60.64	87.86	97.50	74.35	95.55	81.40	79.64
ME(ours)	70.26	82.41	99.33	99.67	91.39	**100.00**	99.01	91.72
BCE	**99.92**	99.75	99.75	**99.75**	97.64	99.92	**99.92**	99.52
PR(ours)	**99.92**	**99.83**	**99.83**	99.50	**98.83**	99.83	**99.92**	**99.67**

Таблица 6: F1 методов детекции на «News Articles», длинные тексты

	GPT-4 Turbo	GPT-4 Omni	Claude3 Opus	Llama3 70B	Cmd R+	YaGPT3 Pro	Giga-Chat	Avg
Proba	43.83	53.55	84.72	86.05	75.25	94.23	86.71	74.91
PPL	53.74	66.70	94.31	95.97	92.56	92.41	68.34	80.58
Rank	62.68	69.38	85.22	84.51	82.43	79.82	40.79	72.12
LRank	46.86	64.76	93.79	95.38	92.09	93.15	66.51	78.93
Entropy	52.88	59.95	93.08	93.18	92.51	89.64	71.74	79.00
GLTR	53.40	63.06	93.06	94.57	89.98	93.21	83.76	81.58
LRR	54.92	56.40	89.76	92.56	88.99	93.12	71.56	78.19
FD-S	61.41	65.50	70.47	81.43	48.51	90.40	48.30	66.57
FD-A	61.36	65.61	70.35	81.29	48.51	90.40	48.35	66.55
ME(ours)	72.51	77.85	95.48	**98.27**	93.60	**99.42**	98.03	90.74
BCE	98.83	98.42	96.27	96.83	95.61	98.52	**98.36**	97.55
PR(ours)	**99.08**	**98.75**	**96.66**	97.24	**95.75**	98.44	**98.36**	**97.75**

Таблица 7: F1 методов детекции на «Creative Writing», длинные тексты

6.2 Cross-Datasource

В данном разделе мы рассматриваем качество моделей при использовании разных источников данных на обучении и тестировании.

На рис. 8 приведены F1 для каждого метода при замене всех трех источников на коротких текстах. На диаграммах явно видим, что в случае fine-tuning методы сильнее переобучаются, то есть F1 преобладают на главной диагонали (in-domain). В других методах это не всегда выполняется и в части замеров качество при изменении домена оказывается в выше.

Агрегированные показатели качества для cross-datasource представлены в табл. 8: здесь мы исключили из усреднения in-domain замеры и усреднили F1 для каждого метода по всем оставшимся источникам данных и всем моделям. На коротких текстах среди отдельных критериев наиболее стабильным оказался FastDetectGPT, в то время как на полных — перплексия.

Самым робастным к изменению датасета (табл. 8) оказался предложенный Metric Ensemble, объединяющий все metric-based признаки.

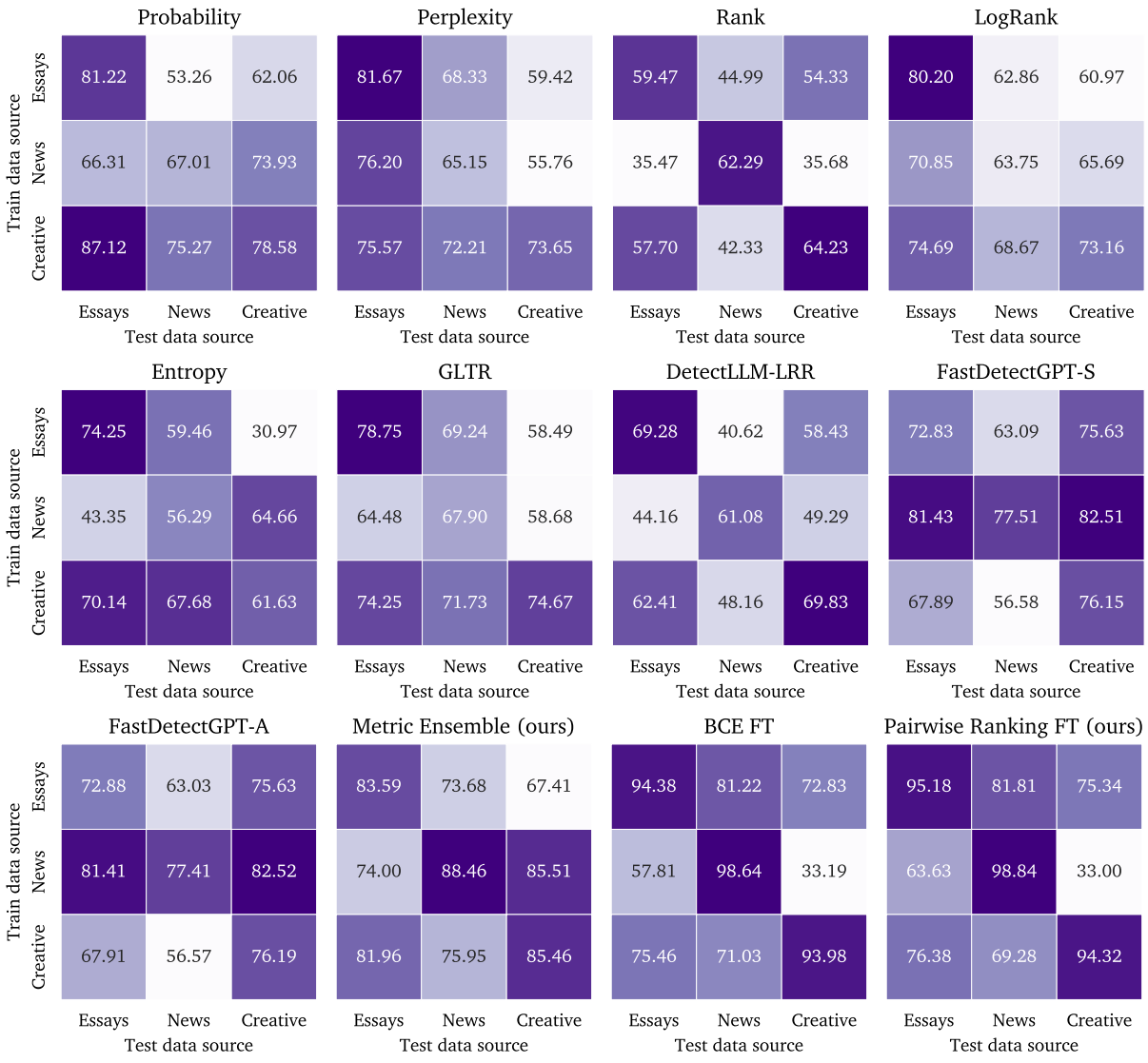

Рис. 8: F1 при обучении и замере на разных источниках данных (короткие тексты), усреднение по всем LLM из разд. 4.2.2

	Короткие тексты	Полные тексты
Probability	69.66	81.25
Perplexity	67.91	81.81
Rank	45.08	59.34
LogRank	67.29	78.13
Entropy	56.04	75.91
GLTR	66.14	77.99
DetectLLM-LRR	50.51	77.15
FastDetectGPT-S	71.19	67.26
FastDetectGPT-A	71.18	67.23
Metric Ensemble (ours)	**76.42**	**82.86**
BCE FT	65.26	80.46
Pairwise Ranking FT (ours)	66.57	81.14

Таблица 8: F1 на out-of-domain источниках данных, в каждом методе результаты усреднены по всем out-of-domain источникам из разд. 4.2.1 и LLM из разд. 4.2.2

6.3 Cross-LLM

В данной серии экспериментов мы исследуем сохранение качества алгоритмов детекции между разными LLM.

На диаграммах рис. 9 приведены F1 на коротких текстах при изменении тестовых LLM. Можно заметить, что достаточно GPT4-модели слабо детектируются всеми методами, кроме fine-tuning. У оставшихся LLM качество чаще переносится между различными методами детекции, при этом встречаются выбросы.

Агрегированные результаты приведены на табл. 9: здесь мы также исключили из усреднения in-domain LLM и усреднили F1 при замере на всех оставшихся, а также усреднили по всем источникам данных. Как видим, в сценарии cross-LLM стабильнее сработали методы, основанные на fine-tuning, причем Pairwise Ranking в среднем достигает более высокого качества.

Наименее надежным на коротких текстах оказался ранговый критерий (Rank) и DetectLLM-LRR, на длинных — Rank и FastDetectGPT.

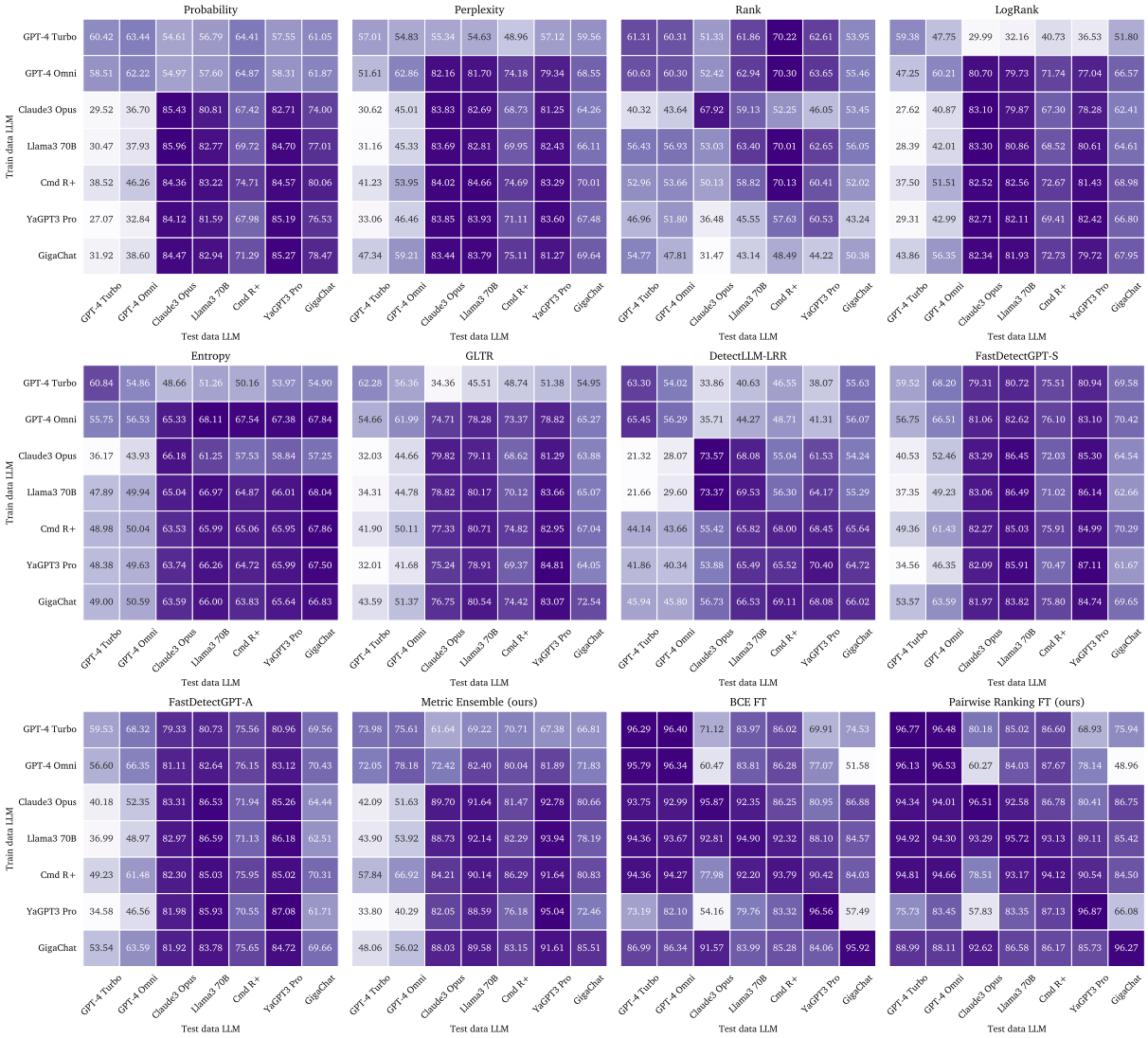

Рис. 9: F1 при обучении и замере на текстах, сгенерированных разными LLM (короткие тексты), усреднение по всем источникам данных из разд. 4.2.1

	Короткие тексты	Полные тексты
Probability	63.16	67.11
Perplexity	65.53	75.28
Rank	53.69	59.27
LogRank	61.44	72.81
Entropy	58.66	73.38
GLTR	62.71	71.61
DetectLLM-LRR	51.81	69.64
FastDetectGPT-S	70.31	61.55
FastDetectGPT-A	70.28	61.53
Metric Ensemble (ours)	73.21	68.93
BCE FT	83.27	89.47
Pairwise Ranking FT (ours)	**84.56**	**90.05**

Таблица 9: F1 на out-of-domain LLM, в каждом методе результаты усреднены по всем источникам из разд. 4.2.1 и out-of-domain LLM из разд. 4.2.2

7 Заключение

В данной работе произведено исследование различных методов детекции текста, сгенерированного большими языковыми моделями. Реализованы современные алгоритмы детекции и произведено их сравнение на различных источниках данных и LLM, на коротких и длинных текстах, а также на out-of-domain данных.

Собран новый бенчмарк для методов детекции, в датасет вошли 7 современных языковых моделей, вышедших в конце 2023 – начале 2024: GPT4 Turbo, GPT4 Omni, Claude 3 Opus, Llama3 70B, CommandR+, YandexGPT 3 Pro, GigaChat Pro. Суммарно для датасета были сгенерировано 21 000 примеров на трех различных доменах. Данный бенчмарк может быть использован в будущих работах для сравнения алгоритмов обнаружения сгенерированного текста или в качестве датасета для обученя.

Показано, что ансамблирование легковесных детекторов позволяет заметно улучшить качество базовых критериев, причем построенный ансамбль является устойчивым к out-of-domain данным.

Предложен новый метод обучения детекторов: Pairwise Ranking Fine-Tuning, основанный на свойстве параллельности датасета и переиспользующий функционал reward-моделей для обучения с подкреплением. Метод улучшает качество стандартного fine-tuning как в in-domain, так и в cross-LLM сценарии.

Тем не менее в Pairwise Ranking Fine-Tuning сохраняется деградация качества при замере на out-of-domain данных. Развитие робастных алгоритмов классификации, качественно работающих на разных доменах, сильно отличающихся от источников в обучающей выборке, является одним из приоритетных направлений будущих исследований методов обнаружения текста, сгенерированного LLM.

Список литературы

1. *Achiam J.*, *Adler S.*, *Agarwal S.*, *Ahmad L.*, *Akkaya I.*, *Aleman F. L.*, *Almeida D.*, *Altenschmidt J.*, *Altman S.*, *Anadkat S.*, [et al.]. Gpt-4 technical report // arXiv preprint arXiv:2303.08774. — 2023.
2. *Team G.*, *Anil R.*, *Borgeaud S.*, *Wu Y.*, *Alayrac J.-B.*, *Yu J.*, *Soricut R.*, *Schalkwyk J.*, *Dai A. M.*, *Hauth A.*, [et al.]. Gemini: a family of highly capable multimodal models // arXiv preprint arXiv:2312.11805. — 2023.
3. *Guo B.*, *Zhang X.*, *Wang Z.*, *Jiang M.*, *Nie J.*, *Ding Y.*, *Yue J.*, *Wu Y.* How close is chatgpt to human experts? comparison corpus, evaluation, and detection // arXiv preprint arXiv:2301.07597. — 2023.
4. *Aich A.*, *Bhattacharya S.*, *Parde N.* Demystifying neural fake news via linguistic feature-based interpretation // Proceedings of the 29th International Conference on Computational Linguistics. — 2022. — P. 6586–6599.
5. *Ayoobi N.*, *Shahriar S.*, *Mukherjee A.* The looming threat of fake and llm-generated linkedin profiles: Challenges and opportunities for detection and prevention // Proceedings of the 34th ACM Conference on Hypertext and Social Media. — 2023. — P. 1–10.
6. *Weidinger L.*, *Mellor J.*, *Rauh M.*, *Griffin C.*, *Uesato J.*, *Huang P.-S.*, *Cheng M.*, *Glaese M.*, *Balle B.*, *Kasirzadeh A.*, [et al.]. Ethical and social risks of harm from language models // arXiv preprint arXiv:2112.04359. — 2021.
7. *Huang L.*, *Yu W.*, *Ma W.*, *Zhong W.*, *Feng Z.*, *Wang H.*, *Chen Q.*, *Peng W.*, *Feng X.*, *Qin B.*, [et al.]. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions // arXiv preprint arXiv:2311.05232. — 2023.
8. *Alemohammad S.*, *Casco-Rodriguez J.*, *Luzi L.*, *Humayun A. I.*, *Babaei H.*, *LeJeune D.*, *Siahkoohi A.*, *Baraniuk R. G.* Self-consuming generative models go mad // arXiv preprint arXiv:2307.01850. — 2023.
9. *Pu J.*, *Sarwar Z.*, *Abdullah S. M.*, *Rehman A.*, *Kim Y.*, *Bhattacharya P.*, *Javed M.*, *Viswanath B.* Deepfake text detection: Limitations and opportunities // 2023 IEEE Symposium on Security and Privacy (SP). — IEEE. 2023. — P. 1613–1630.
10. *Solaiman I.*, *Brundage M.*, *Clark J.*, *Askell A.*, *Herbert-Voss A.*, *Wu J.*, *Radford A.*, *Krueger G.*, *Kim J. W.*, *Kreps S.*, [et al.]. Release strategies and the social impacts of language models // arXiv preprint arXiv:1908.09203. — 2019.
11. *Mitchell E.*, *Lee Y.*, *Khazatsky A.*, *Manning C. D.*, *Finn C.* Detectgpt: Zero-shot machine-generated text detection using probability curvature // International Conference on Machine Learning. — PMLR. 2023. — P. 24950–24962.
12. *Wu J.*, *Yang S.*, *Zhan R.*, *Yuan Y.*, *Wong D. F.*, *Chao L. S.* A Survey on LLM-gernerated Text Detection: Necessity, Methods, and Future Directions // CoRR. — 2023. — Vol. abs/2310.14724. — arXiv: 2310.14724.
13. *Verma V.*, *Fleisig E.*, *Tomlin N.*, *Klein D.* Ghostbuster: Detecting text ghostwritten by large language models // arXiv preprint arXiv:2305.15047. — 2023.
14. *Gehrmann S.*, *Strobelt H.*, *Rush A. M.* Gltr: Statistical detection and visualization of generated text // arXiv preprint arXiv:1906.04043. — 2019.

15. *Radford A.*, *Wu J.*, *Child R.*, *Luan D.*, *Amodei D.*, *Sutskever I.*, [et al.]. Language models are unsupervised multitask learners // OpenAI blog. — 2019. — Vol. 1, no. 8. — P. 9.

16. *He X.*, *Shen X.*, *Chen Z.*, *Backes M.*, *Zhang Y.* Mgtbench: Benchmarking machine-generated text detection // arXiv preprint arXiv:2303.14822. — 2023.

17. *Liu Y.*, *Zhang Z.*, *Zhang W.*, *Yue S.*, *Zhao X.*, *Cheng X.*, *Zhang Y.*, *Hu H.* Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models // arXiv preprint arXiv:2304.07666. — 2023.

18. *Weber-Wulff D.*, *Anohina-Naumeca A.*, *Bjelobaba S.*, *Foltýnek T.*, *Guerrero-Dib J.*, *Popoola O.*, *Šigut P.*, *Waddington L.* Testing of detection tools for AI-generated text // International Journal for Educational Integrity. — 2023. — Vol. 19, no. 1. — P. 26.

19. *Akram A.* An Empirical Study of AI Generated Text Detection Tools // arXiv preprint arXiv:2310.01423. — 2023.

20. *King J.*, *Baffour P.*, *Crossley S.*, *Holbrook R.*, *Demkin M.* LLM - Detect AI Generated Text. — 2023. — URL: `https://kaggle.com/competitions/llm-detect-ai-generated-text`.

21. *Uchendu A.*, *Ma Z.*, *Le T.*, *Zhang R.*, *Lee D.* Turingbench: A benchmark environment for turing test in the age of neural text generation // arXiv preprint arXiv:2109.13296. — 2021.

22. *Macko D.*, *Moro R.*, *Uchendu A.*, *Lucas J. S.*, *Yamashita M.*, *Pikuliak M.*, *Srba I.*, *Le T.*, *Lee D.*, *Simko J.*, [et al.]. MULTITuDE: Large-Scale Multilingual Machine-Generated Text Detection Benchmark // arXiv preprint arXiv:2310.13606. — 2023.

23. *Koike R.*, *Kaneko M.*, *Okazaki N.* Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples // Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 38. — 2024. — P. 21258–21266.

24. *Su Z.*, *Wu X.*, *Zhou W.*, *Ma G.*, *Hu S.* Hc3 plus: A semantic-invariant human chatgpt comparison corpus // arXiv preprint arXiv:2309.02731. — 2023.

25. *Wang Y.*, *Mansurov J.*, *Ivanov P.*, *Su J.*, *Shelmanov A.*, *Tsvigun A.*, *Whitehouse C.*, *Afzal O. M.*, *Mahmoud T.*, *Aji A. F.*, [et al.]. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection // arXiv preprint arXiv:2305.14902. — 2023.

26. *Houvardas J.*, *Stamatatos E.* N-gram feature selection for authorship identification // International conference on artificial intelligence: Methodology, systems, and applications. — Springer. 2006. — P. 77–86.

27. *Chiang W.-L.*, *Zheng L.*, *Sheng Y.*, *Angelopoulos A. N.*, *Li T.*, *Li D.*, *Zhang H.*, *Zhu B.*, *Jordan M.*, *Gonzalez J. E.*, [et al.]. Chatbot arena: An open platform for evaluating llms by human preference // arXiv preprint arXiv:2403.04132. — 2024.

28. *Holtzman A.*, *Buys J.*, *Du L.*, *Forbes M.*, *Choi Y.* The curious case of neural text degeneration // arXiv preprint arXiv:1904.09751. — 2019.

29. *Team G.*, *Mesnard T.*, *Hardin C.*, *Dadashi R.*, *Bhupatiraju S.*, *Pathak S.*, *Sifre L.*, *Rivière M.*, *Kale M. S.*, *Love J.*, [et al.]. Gemma: Open models based on gemini research and technology // arXiv preprint arXiv:2403.08295. — 2024.

30. *Raffel C.*, *Shazeer N.*, *Roberts A.*, *Lee K.*, *Narang S.*, *Matena M.*, *Zhou Y.*, *Li W.*, *Liu P. J.* Exploring the limits of transfer learning with a unified text-to-text transformer // Journal of machine learning research. — 2020. — Vol. 21, no. 140. — P. 1–67.

31. *Su J.*, *Zhuo T. Y.*, *Wang D.*, *Nakov P.* Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text // arXiv preprint arXiv:2306.05540. — 2023.

32. *Bao G.*, *Zhao Y.*, *Teng Z.*, *Yang L.*, *Zhang Y.* Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature // arXiv preprint arXiv:2310.05130. — 2023.

33. *Wang A.*, *Singh A.*, *Michael J.*, *Hill F.*, *Levy O.*, *Bowman S. R.* GLUE: A multi-task benchmark and analysis platform for natural language understanding // arXiv preprint arXiv:1804.07461. — 2018.

34. *Wang A.*, *Pruksachatkun Y.*, *Nangia N.*, *Singh A.*, *Michael J.*, *Hill F.*, *Levy O.*, *Bowman S.* Superglue: A stickier benchmark for general-purpose language understanding systems // Advances in neural information processing systems. — 2019. — Vol. 32.

35. *Stiennon N.*, *Ouyang L.*, *Wu J.*, *Ziegler D.*, *Lowe R.*, *Voss C.*, *Radford A.*, *Amodei D.*, *Christiano P. F.* Learning to summarize with human feedback // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 3008–3021.

36. *Ouyang L.*, *Wu J.*, *Jiang X.*, *Almeida D.*, *Wainwright C.*, *Mishkin P.*, *Zhang C.*, *Agarwal S.*, *Slama K.*, *Ray A.*, [et al.]. Training language models to follow instructions with human feedback // Advances in neural information processing systems. — 2022. — Vol. 35. — P. 27730–27744.

37. *Sanh V.*, *Debut L.*, *Chaumond J.*, *Wolf T.* DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter // ArXiv. — 2019. — Vol. abs/1910.01108.