ЗАДАЧА ДЕТЕКЦИИ ГРАНИЦЫ МЕЖДУ ЧЕЛОВЕЧЕСКИМ И МАШИННО-ГЕНЕРИРУЕМЫМ КОНТЕНТОМ

Г. В. Пшеничников[1, *] А. В. Грабовой[2, **]

[1] *119234, Москва, территория Ленинские Горы, 1с52, МГУ имени М.В.Ломоносова, 2-й учебный корпус, Россия*

[2] *141701 Долгопрудный, М.о., Институтский пер., 9, МФТИ, Россия*

Целью данного исследования является разработка метода определения номера предложения, с которого начинается сгенерированная часть гибридного текста. Алгоритм основан на перплексии предложений, вычисленной при помощи собранных словарей вероятностей, которая вместе со структурными и стилистическими характеристиками подается на вход бинарному классификатору, определяющему сгенерировано ли предложение, и после анализа полученных меток текст делится на две части. Исследование подтверждает гипотезу о том, что перплексия зависит от типа текста, сгенерирован он или нет, при смене типа происходит резкое изменение ее значения — признак, по которому находится искомая граница перехода. Модель достигает высокой точности на текстах, содержащих не более 12-и предложений, алгоритм ошибается не более чем на 1 предложение; на более длинных текстах, содержащих не более 60-и предложений, алгоритм ошибается не более чем на 10 предложений. При этом классификатор, используемый в методе, достигает в среднем 80% точности и 80% полноты в обнаружении сгенерированного предложения.

Ключевые слова: детекция машинного-сгенерированного текста, перплексия, большие языковые модели, гибридный текст, бинарная классификация.

* Electronic address: glebpsh@mail.ru

** Electronic address: grabovoy.av@phystech.edu

1. ВВЕДЕНИЕ

С развитием больших языковых моделей — Large Language Models (LLM) решение задач, таких как детекции целых текстов, отдельных фрагментов, написанных искусственным интеллектом, становится все более актуальной. Языковые модели способны создавать связные, осмысленные тексты, которые сложно отличить от текстов, написанных человеком [1–3]. Активно развиваются нейронные сети, такие как ChatGPT [4], Llama [5], Mistral [6] и другие, помогающие в написании научных статей, что приводит к увеличению количества машинных генераций в данных работах, к стремительному росту академического плагиата [7, 8]. Также стоит проблема неточности информации, создаваемой языковыми моделями, что может привести к дезинформации пользователей [9, 10]. В отдельную категорию задач, требующих решения, можно вынести анализ гибридных текстов, созданных человеком и искусственным интеллектом совместно. Эта задача занимает важное место, так как такие тексты отражают реальные сценарии использования генеративных моделей для дополнения, расширения статей, научных работ, образовательных текстов [8, 11].

Цель данной работы заключается в решении этой проблемы — определение точки перехода, на уровне предложений, в которой текст, написанный человеком, сменяется машинной генерацией. На данный момент существуют сервисы, частично решающие поставленную задачу, например zero-shot DetectGPT [12], Binoculars [13], FastDetectGPT и DNA-GPT [14]. Однако, данные программы требуют значительных вычислительных ресурсов, так как разнообразие способов генерации у языковых моделей растет [15, 16]. Метод, предложенный в данной статье, лишен этого недостатка и способен определять начало сгенерированного фрагмента даже у закрытых генеративных моделей с недоступными весами.

Исследуемый метод основывается на вычислении перплексии и логарифмической функции правдоподобия [12, 16]. Выбор данного признака обусловлен тем, что генеративные модели обучаются создавать тексты, минимизирующие уровень перплексии [17, 18], поэтому, анализируя данную характеристику текста, можно судить о сгенерированности. Предложенная модель использует статистические словари, создание которых происходит на основе набора сгенерированных текстов, что является преимуществом данного метода, так как тексты можно сгенерировать без доступа к весам LLM. На

основе словарей вычисляются перплексии предложений анализируемого текста, которые подаются на вход бинарному классификатору. Итоговая граница определяется при помощи Индекса Джини.

Предложенный метод протестирован на наборах тестовых данных с соревнований SemEval-2024 Task 8 [19], DAGPap24 [20], COLING 2025 [21], PAN 2024 [22] и PAN 2025 [23] и показал высокую точность в классификации предложений, в определении оптимальной границы перехода на машинную генерацию. На текстах с небольшим числом предложений — до 10, классификатор достигает 93.2% полноты и 91.1% точности, при этом ошибка в определении границы не более 1-го предложения. Для текстов со средней длиной 32 предложения классификатор показывает полноту 92.93% и точность 70.7%, средняя ошибка в определении номера предложения перехода — 9.2 предложений. Также определено качество метода для длинных текстов со средней длиной от 40 до 60 предложений: полнота — 75.9%, точность — 78.2%, средняя ошибка — 9.6 предложений. Основной вклад данной статьи заключается в разработке метода разделения текста на вышеупомянутые части, в разработке алгоритма анализа перплексии токенов и предложений на основе словарей вероятностей.

2. СМЕЖНЫЕ ИССЛЕДОВАНИЯ

Проблемы использования машинной генерации при создании научных работ и статей, получения некорректной информации [24], в последствии используемой пользователями интернета, всегда стояло остро. Область создания антиплагиатов, детекторов машинной генерации развивается на протяжении многих лет [25]. Использование численных характеристик текста, таких как GROVER, N-gram, рукотворных признаков для анализа текстов практикуется давно [15].

Zero-shot DetectGPT [12] анализирует кривизну функции логарифмической вероятности, сравнивая оригинальный текст с его возмущенными версиями. Метод Binoculars [13] основан на сравнении перплексии двух близкородственных языковых моделей. FastDetectGPT использует условную кривизну вероятности токенов. DNA-GPT [14] регенерирует часть текста и сравнивает ее с оригинальной частью через n-граммы. Эти подходы требуют значительных вычислительных ресурсов, доступ к весам генеративных моделей, также эти методы имеют ограниченную универсальность при работе с

новыми архитектурами LLM.

Использование перплексии для выявления сгенерированных текстов или его фрагментов имеет место и сейчас. Активно развиваются подходы обнаружения машинной генерации [26, 27], основанные на анализе этой характеристики текста. В качестве основы для исследуемого подхода взяты методы из статей [26, 27].

3. ПОСТАНОВКА ЗАДАЧИ

Пусть W — алфавит токенов, минимальных элементов текста. В качестве входных данных имеем набор текстов, состоящих из предложений:

$$D = \left\{ [t_j]_{j=1}^{n} \mid t_j \in W, n \in \mathbb{N} \right\}.$$

Имеется набор текстов $T = \{t^j\}_{j=1}^{M}$, где каждый текст представляет собой последовательность предложений $t^j = \{s_1^j, s_2^j, \ldots, s_{m^j}^j\}$. Требуется найти такие номера i^j для $j = 1, \ldots, M$, что $\{s_1^j, s_2^j, \ldots, s_{i^j-1}^j\}$ — текст, написанный человеком, а $\{s_{i^j}^j, s_{i^j+1}^j, \ldots, s_{m^j}^j\}$ — текст, написанный при помощи искусственного интеллекта.

Требуется разработать модель:

$$a = \arg\min_{a^* \in A} \mathcal{L}(I_{\text{true}}, I_{\text{pred}}), \tag{1}$$

$$\mathcal{L}(I_{\text{true}}, I_{\text{pred}}) = \frac{1}{M} \sum_{j=1}^{M} |i_{\text{true}}^j - i_{\text{pred}}^j| \tag{2}$$

где A — множество моделей, решающих данную задачу, $I_{true} = \{i_{\text{true}}^j\}_{j=1}^{M}, I_{pred} = \{i_{\text{pred}}^j\}_{j=1}^{M}$ — векторы истинных и предсказанных номеров предложений, в которых происходит переход между частями текстов, $\mathcal{L}$ — функция потерь, используется Mean Absolute Error, качество модели будет оцениваться по этой метрике.

4. МЕТОДОЛОГИЯ

Общий ход метода описан на рис. 1. Вспомогательными данными, необходимые для создания словарей вероятностей, являются наборы текстов, созданные при помощи генеративных языковых моделей.

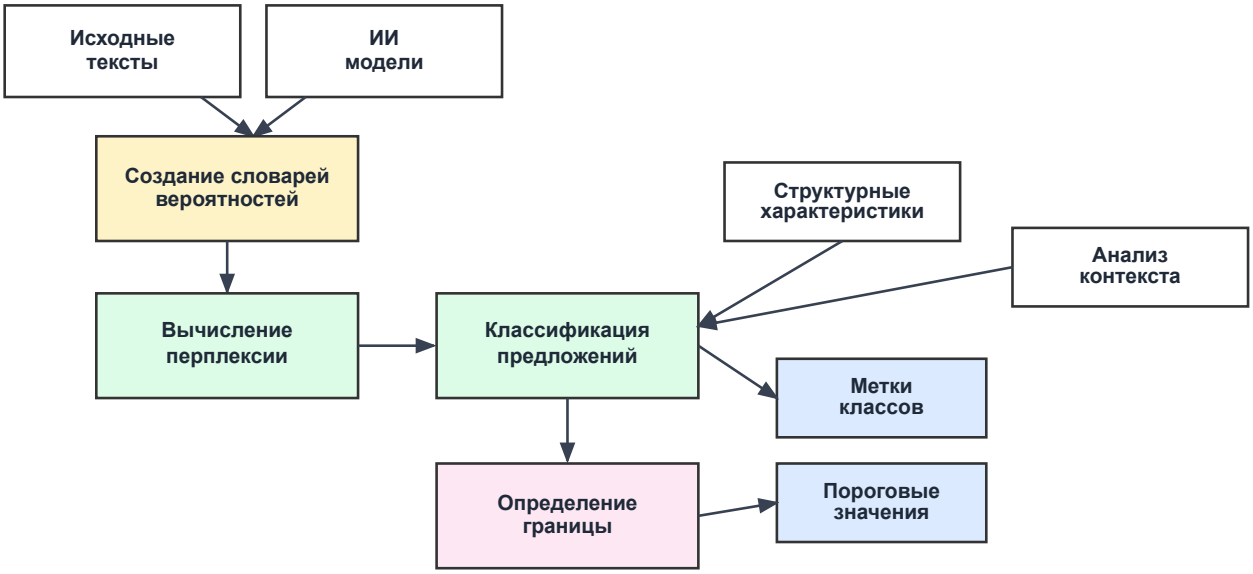

Рис. 1. Общий ход метода

4.1. Создание словарей вероятностей

Словарь вероятностей — структура данных, хранящая токены и соответствующие им распределения вероятностей последующих токенов. Словарь необходим для подсчета перплексии.

Словари созданы при помощи нескольких генеративных моделей: GPT-2 [28], BART [29], BLOOM [30] и OPT [31]. Выбиралось 1000 случайных промптов из датасета HC3 [32] из раздела all, и на их основе генерировались тексты. Далее выбирались наиболее часто встречающиеся n-граммы ($n = 1, 2, 3$), для них модели предсказывали вероятности следующего токена. В словарь занесены наборы из токены, имеющих наибольшие вероятности. В итоге создано три словаря: с униграммами, с биграммами и с триграммами.

4.2. Вычисление перплексии предложений текстов

Ключевой аспект всего алгоритма — вычисление перплексии. Перплексия — это мера неопределенности модели при предсказании токенов, чем меньше перплексия, тем увереннее модель в своих предсказаниях — исходя из этого выдвинута гипотеза, что

на основе перплексии предложений можно найти место, где текст становится сгенерированным. Для этого использовались все три словаря вероятностей: для определения локальных закономерностей — уровень слов, и для определения контекстных зависимостей — уровень фраз, словосочетаний.

Перплексия (PP) вычисляется отдельно для каждого типа n-грамм. Общая формула имеет вид:

$$PP_n(S) = \exp\left(-\frac{1}{N-n+1}\sum_{i=n}^{N}\log P(w_i|w_{i-n+1},\ldots,w_{i-1})\right), \tag{3}$$

где $S = \{w_1, w_2, \ldots, w_N\}$ — предложение, последовательность токенов, $P(w_i|w_{i-n+1},\ldots,w_{i-1})$ — вероятность того, что после последовательности токенов $w_{i-n+1},\ldots,w_{i-1}$ будет идти токен w_i, N — количество токенов в предложении, n — размер n-граммы. Эти вероятности мы получаем из словарей.

Для униграмм ($n = 1$) формула имеет вид:

$$PP_1(S) = \exp\left(-\frac{1}{N-1}\sum_{i=1}^{N-1}\log P(w_{i+1}|w_i)\right), \tag{4}$$

где $P(w_{i+1}|w_i)$ — вероятность токена w_{i+1} при условии предыдущего токена w_i.

Для биграмм ($n = 2$) перплексия вычисляется как:

$$PP_2(S) = \exp\left(-\frac{1}{N-2}\sum_{i=1}^{N-2}\log P(w_{i+2}|w_i, w_{i+1})\right), \tag{5}$$

где $P(w_{i+2}|w_i, w_{i+1})$ — вероятность токена w_{i+2} при условии двух предыдущих токенов w_i и w_{i+1}.

Для триграмм ($n = 3$) формула принимает вид:

$$PP_3(S) = \exp\left(-\frac{1}{N-3}\sum_{i=1}^{N-3}\log P(w_{i+3}|w_i, w_{i+1}, w_{i+2})\right), \tag{6}$$

где $P(w_{i+3}|w_i, w_{i+1}, w_{i+2})$ — вероятность токена w_{i+3} при условии трех предыдущих токенов w_i, w_{i+1} и w_{i+2}.

При отсутствии n-граммы в словарях — пропускаем ее, не учитываем при расчетах. Если в предложении нет ни одной n-граммы из словарей, то считаем, что его перплексия большая. Если найдена n-грамма в словаре, то анализируем следующий после нее токен. Если он есть в словаре для соответствующей n-граммы, то его вероятность найдена.

Если нет — используем сглаживание с помощью равномерного распределения между всеми не вошедшими в словарь токенами оставшейся вероятности:

$$P_{\text{smooth}} = \frac{1 - \sum_{j=1}^{k} P(q_j | w_i, \ldots, w_{i+n-1})}{V - k}, \tag{7}$$

где k — количество n-грамм, вошедших в словарь, V — общее количество всевозможных n-грамм, q_j — вероятность токена после n-граммы, который находится в словаре.

4.3. Классификация

После предыдущего шага получены наборы перплексий предложений, составляющих текст. Они необходимы для формирования признаков, передающихся бинарному классификатору. В данном исследовании использовался классификатор на основе случайного леса. Он анализирует каждое предложение текста и генерирует одну из двух возможных меток: класс 0 означает, что данное предложение написано человеком, а класс 1 — что предложение сгенерировано.

В качестве признаков, на которых обучался классификатор, использовался набор из 21-й характеристик текста. Признаки на основе перплексии включают: перплексию каждого предложения для униграмм, вычисляемую по формуле (4), перплексию для биграмм по формуле (5), и перплексию для триграмм по формуле (6). Дополнительно для каждого типа n-грамм вычисляются производные признаки: изменение перплексии как разность между перплексией текущего и предыдущего предложений, локальная средняя перплексия как среднее значение перплексии в окне ± 1 предложение, и отклонение от локальной как разность между перплексией предложения и локальной средней. Лингвистические признаки состоят из количества знаков препинания, заглавных букв и цифр, бинарных признаков начала предложения с заглавной буквы и окончания точкой. Структурные признаки содержат количество слов в каждом предложении и количество символов в нем, относительную позицию предложения в тексте и среднюю длину слов в символах в каждом отдельном предложении.

4.4. Определение оптимальной границы

После классификации всех предложений необходимо найти оптимальное разделение — реализовано при помощи минимизации Индекса Джини. Он определяется фор-

мулой:

$$G(T) = 1 - \sum_{i=0}^{1} p_i^2 = 1 - (p_0^2 + p_1^2), \tag{8}$$

где T — набор предложений (текст), p_0 — доля предложений класса 0 (написано человеком), p_1 — доля предложений класса 1 (сгенерировано).

Для выбора границы минимизируется взвешенный Индекс Джини:

$$G_{\text{weighted}}(i) = \frac{|S_{\text{left}}|}{|S|} \cdot G(S_{\text{left}}) + \frac{|S_{\text{right}}|}{|S|} \cdot G(S_{\text{right}}), \tag{9}$$

где $|S|$ — общее количество предложений, S_{left} — набор предложений до границы i, S_{right} — набор предложений после границы i, $G(S_{\text{left}})$ и $G(S_{\text{right}})$ — индексы Джини для левой и правой частей соответственно.

Находится оптимальная граница:

$$i^* = \arg\min_{i \in N} G_{\text{weighted}}(i), \tag{10}$$

где N — множество позиций, в которых может находиться граница. N определяется тем условием, что после выбранной позиции обязательно должно идти R идущих подряд сгенерированных предложений, в разработанной модели их количество равно 2. В случае, если множество N пусто, то границей считаем номер первого предложения из подпоследовательности предложений длины R, имеющей максимальную суммарную вероятность метки 1.

5. ЭКСПЕРИМЕНТЫ

Для проведения экспериментов использовался набор тестовых данных SemEval-2024 Task 8, подзадача C, так как в данном соревновании решается задача определения оптимальной границы разделения на человеческий и сгенерированный — именно эту задачу и решает реализованный в данной статье метод. В рассматриваемом примере текста метод правильно определил границу перехода.

На рис. 2 видно, что перплексия текста нестабильна, и ее действительно можно использовать в качестве признаков для обучения классификатора. У первых 80-и токенов перплексия почти не меняется, график скользящего окна почти горизонтален — это часть текста написана человеком. После этого токена график резко идет вниз, перплексия стремительно уменьшается, это говорит о том, что генеративная языковая модель с

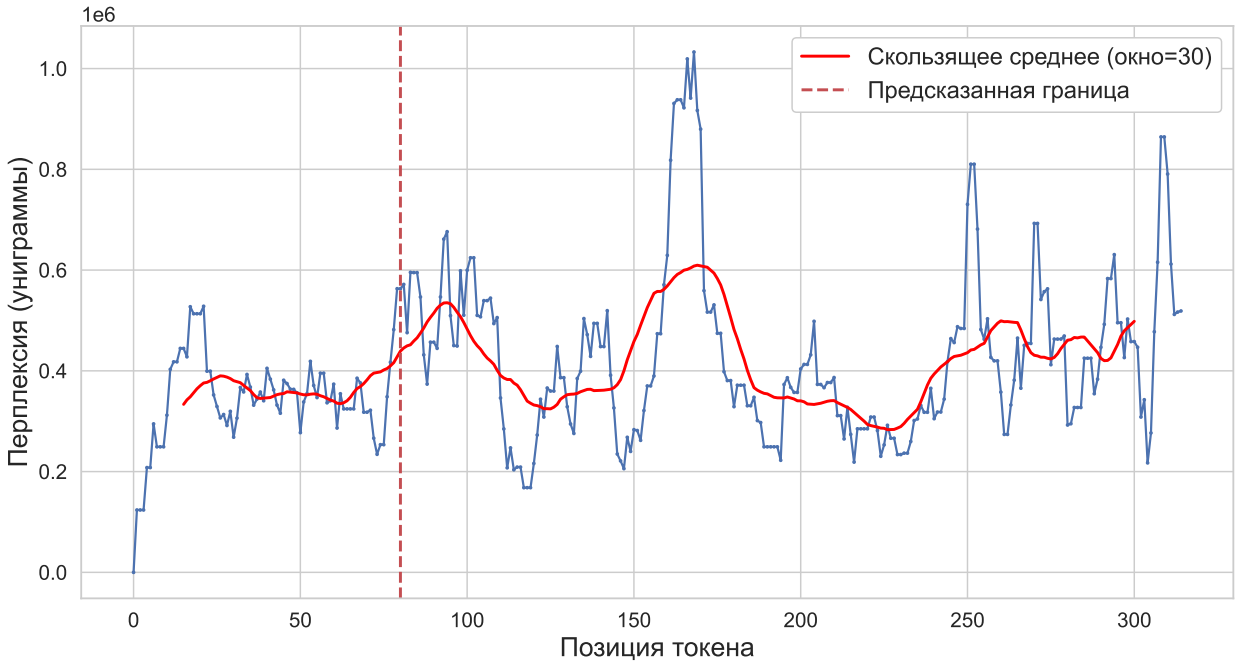

Рис. 2. Изменение перплексии по токенам в смешанном тексте.

большей вероятностью предскажет следующие токены. Начало участка быстрого уменьшения перплексии — начало сгенерированной части. Дальнейшие скачки возникают из-за ограниченности словарей вероятностей — при увеличении их мощности, перплексия будет стабильнее. Гипотеза о том, что при смене авторства с человека на искусственный интеллект происходит резкое уменьшение значения перплексии, подтверждается.

Также рассмотрим изменение перплексии в тексте на уровне предложений на рис. 3. Видно — гипотеза так же подтверждается: по резкому уменьшению перплексии можно судить о границе перехода между частями текста.

6. РЕЗУЛЬТАТЫ

Тестирование модели проводилось на данных с SemEval-2024 Task 8 подзадача C [18]. При анализе результатов на уровне предложений получены следующие значения: средняя ошибка составила 1,03 предложения, медианная ошибка — 1,00 предложение, предсказывание правильного номера предложения достигнуто для 38.9% всего тестового датасета, ошибка в одно предложение получена для 77.2% всего тестового датасета.

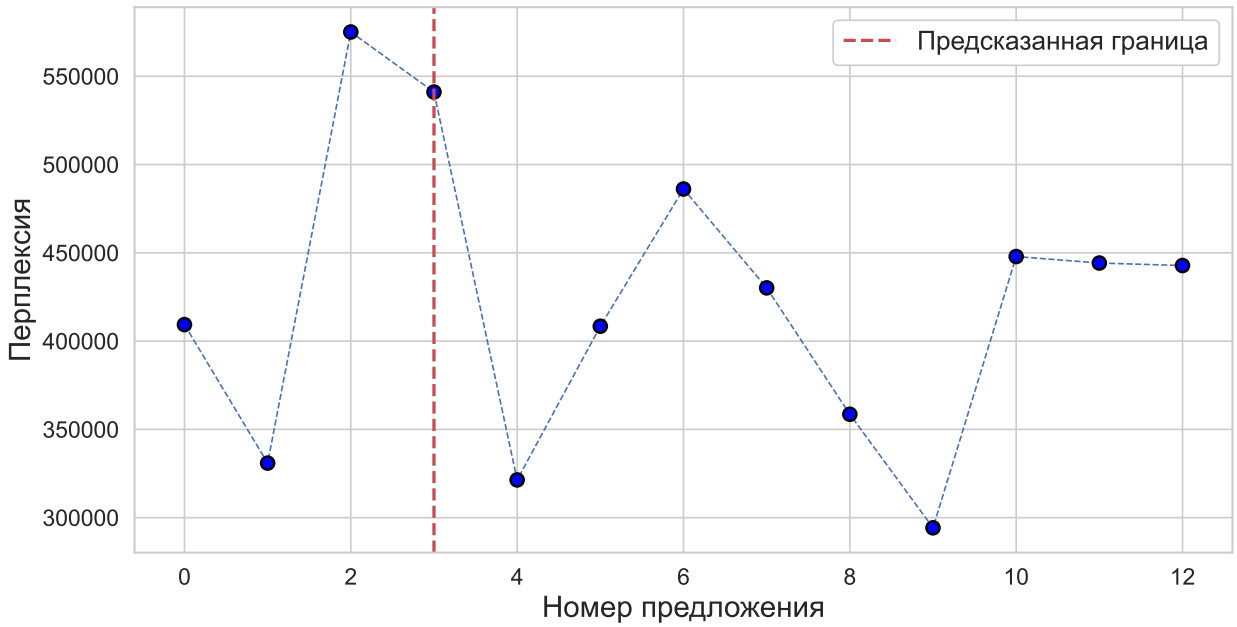

Рис. 3. Изменение перплексии по предложениям в смешанном тексте.

Также в качестве оценки классификации предложений, третий шаг алгоритма, вычислены полнота 93.2% и точность 91.1%. Получены высокие значения, это говорит о том, что классификатор, используя перплексию предложений и структурные характеристики, действительно хорошо обнаруживает сгенерированные предложения, редко их пропускает. Тем самым еще раз подтверждается гипотеза о том, что перплексия зависит от типа написания текста и ее использование в решении задачи, поставленной в рамках данной статьи, оправданно.

Также модель протестирована на данных с DAGPap24 [32]. В данном соревновании решалась задача разделения текста на фрагменты, каждый из которых либо написан человеком, либо модифицирован путем замены слов синонимами с помощью библиотеки NLTK, либо сгенерирован ChatGPT, либо получен путем сокращения более длинного текста. Для исследования результатов модели данной статьи из всех текстов выделены фрагменты, в которых за человеческим текстом следует сгенерированный. Средняя длина полученных фрагментов составляет 59 предложений. Модель показала хорошие результаты — средняя ошибка в определении целевого номера предложения составила 10. При этом классификатор предложений показал высокое качество: полнота 99.4% и точность 82.7%.

Алгоритм был также проверен на выборке с COLING 2025 [19]. На данном соревновании предоставлены тексты, полностью написанные человеком и полностью сгенерированные LLM. Это тексты были соединены — к тексту, написанному человеком присоединялся текст, созданный искусственным интеллектом. Таким образом получили тестовую выборку для исследования в нужном формате. Созданные тексты имели среднюю длину в 31 предложение. На данном выборке модель ошибалась в среднем на 5.5 предложений в определении границы, при этом классификатор показал точность 82.1% и полноту предсказаний 93.6%. Тем самым можно судить о высокой точности определения границы и классификации разработанного метода.

Таблица 1. Результаты тестирования модели на текстах, сгенерированных различными языковыми моделями. Приведены средняя длина текста в предложениях, средняя ошибка определения границы в предложениях, полнота и точность классификации предложений.

Модель	**Средняя длина**	**Средняя ошибка**	**Полнота (%)**	**Точность (%)**
Text-Bison-002	45.7	9	70	80.4
Alpaca-7B	31.7	10.9	94.8	65.5
GPT2-Open-Instruct-V1	43.4	9.9	80.7	77.2
Mistral-7B-Instruct-V0.2	48.7	10.7	70.9	78.1
Llama-2-7B-Chat	41.7	9	75.2	78.5
GPT-3.5-Turbo-0125	38.5	7.1	69.7	81.4
GPT-4-Turbo-Preview	44	8.7	68.5	80.1
BLOOMZ-7B1	38.9	11.3	89	70.8
Gemini-Pro	46.9	10.5	71.3	77.6
Qwen1.5-72B-Chat	40.5	8.5	73	78.9
Llama-2-70B-Chat	45.9	9.7	72.8	78.8
Mixtral-8x7B-Instruct	46.5	9.8	70.5	78.8
Alpaca-13B	31	11.3	90.4	63.4

В качестве дополнительной проверки алгоритма взяты выборки с соревнований PAN 2024 [22] и PAN 2025 task 1 [23]. На этих соревнованиях, так же, как и на COLING

2025 [19], решалась задача бинарной классификации человеческих и сгенерированных текстов. Поэтому была создана новая выборка путем соединения двух типов текстов. Модель была дообучена на тренировочном датасете, созданном при помощи выборок с этих соревнований. На PAN 2024 [22] сгенерированные тексты созданы разными LLM. Результаты тестирования представлены в табл. 1. Метод показывает хорошие результаты, эффективно работая с текстами разных языковых моделей. Наименьшую ошибку метод показал при определении границы в текстах, сгенерированная часть которых создана при помощи моделей GPT семейства. Также метод показал высокую полноту классификации для модели Alpaca — более 90%. Важным наблюдением является тот факт, что статистические словари, при помощи которых создавалась перплексия, были созданы при помощи лишь 4-х генеративных моделей: GPT-2 [28], BART [29], BLOOM [30] и OPT [31], но при этом метод эффективно определяет начало фрагмента, созданных 13-ю языковыми моделями. Это доказывает универсальность подхода. Тестирование также проведено на текстах из тестовой выборки PAN 2025 task 1 [23]. Результаты высоки: при средней длине текстов 39 предложения метод ошибался не более, чем на 10 предложений; классификатор показал полноту 75.9% и точность 73.8%.

Проведено сопоставление алгоритма и онлайн сервисов GPTZero [33], Moxby [34], Copyleaks [35], находящих сгенерированные предложения и целевой номер. Сравнение проводилось по значению ошибки в определении предложения перехода и по времени, необходимого для анализа текста. Для сопоставления методов использовалось два набора текстов — со средней длиной в 13.1 предложения и в 43.6 предложения. Результаты представлены в табл. 2 и табл. 3 соответсвенно.

Таблица 2. Сравнение алгоритма и онлайн сервисами по значению средней ошибки в предложениях и по времени. Средняя длина текстов 13.1 предложения.

	Метод статьи	**GPTZero**	**Moxby**	**Copyleaks**
Средняя ошибка	1.0	0.9	1.4	4.0
Время, с	0.033	1.968	7.287	1.210

Для текстов с длиной менее 13.1 предложений метод показывает ошибку, которая меньше, чем у сервисов Moxby и Copyleaks, и сравнима с ошибкой GPTZero. При этом для метода статьи требуется значительно меньше времени для анализа текста.

Таблица 3. Сравнение алгоритма и онлайн сервисами по значению средней ошибки в предложениях и по времени. Средняя длина текстов 43.6 предложения.

	Метод статьи	**GPTZero**	**Moxby**	**Copyleaks**
Средняя ошибка	7.9	4.0	7.7	7.3
Время, с	0.044	4.541	8.234	2.540

Для текстов с длиной не более 43.6 предложения метод уступают сервису GPTZero по значению ошибки, но в то же время показывает сравнимую с Moxby и Copyleaks ошибку. Превосходно метода статьи по времени увеличилось, алгоритму необходимо в 100 раз меньше времени, чем онлайн сервисам.

Итого, можно сделать вывод, что алгоритм данной статьи является оптимальным решением поставленной задачи, показывает ошибку, сопоставимую с ошибкой имеющихся онлайн сервисов, но занимает значительно меньшее время.

7. АНАЛИЗ ОШИБОК

На рис. 4 наблюдается положительная корреляция между процентом ошибочных предсказаний и длиной текста. Линия тренда демонстрирует четкий рост процента ошибок при увеличении количества предложений в тексте. Для коротких текстов, до 10 предложений, процент ошибок составляет 30–75%, при этом наблюдается значительный разброс значений. При количестве предложений от 3 до 10 процент ошибок резко возрастает и держится примерно на одном уровне: 60–75% ошибок. При увеличении длин текстов до 75 предложений процент ошибок стабильно растет до 85–95%. Для длинных текстов более 75 предложений процент ошибочных предсказаний около 100%, что указывает на невозможность применения метода для длинных текстов.

Анализируя рис. 5, видно, что модель эффективна для текстов с небольшой длиной, до 60-и предложений. Для текстов длиной до 60-и предложений средняя ошибка меньше 10-и, однако с увеличением длины текста до 100-а предложений ошибка растет линейно и доходит до 40-а. Далее, с увеличением количества предложений в тексте, ошибка растет нелинейно, имеет место большая вариативность значений.

По полученным результатам можно судить о высокой точности предсказаний гра-

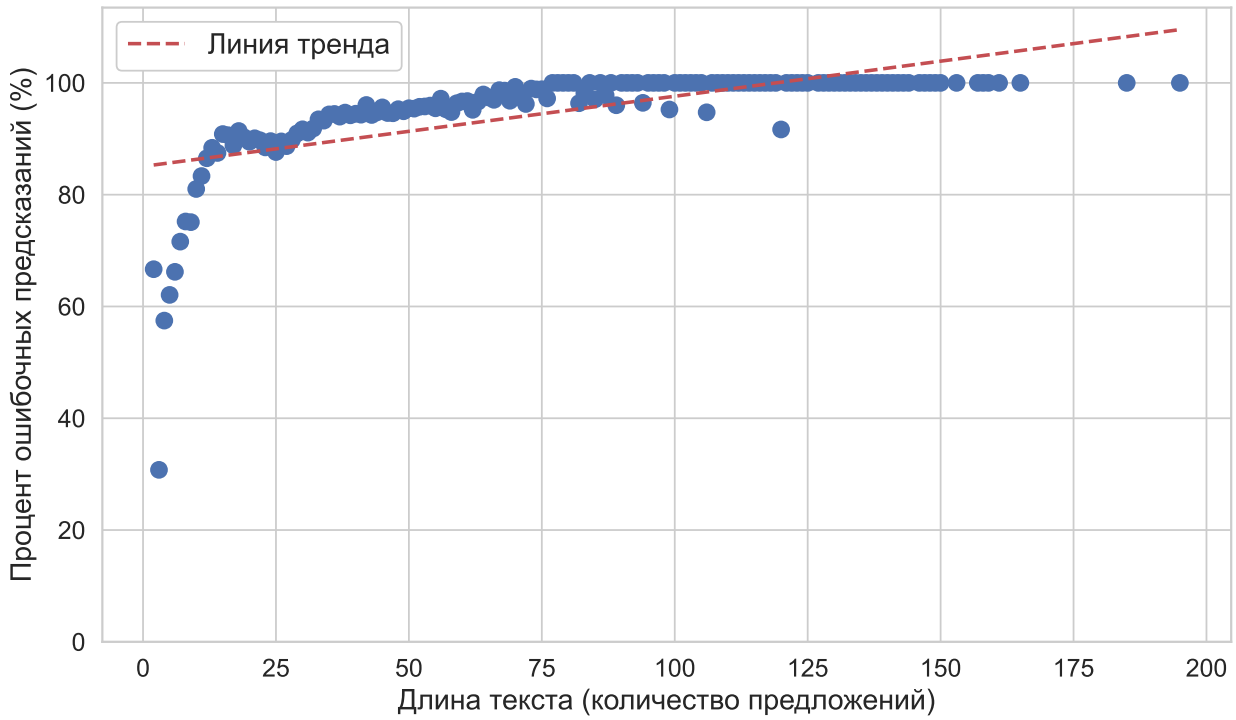

Рис. 4. Зависимость процента ошибочных предсказаний от длины текста.

ницы для текстов до 60-и предложений. Для текстов с большим числом предложений метод часто ошибается и предсказывает границы, сильно отличающиеся от истинных. Это объясняется тем, что словари вероятностей содержат недостаточное количество токенов, и поэтому в длинных предложениях содержится большое количество мест с резким изменением перплексии, которые могут быть расценены как граница перехода. Это делает метод нестабильным.

8. ЗАКЛЮЧЕНИЕ

В данной статье реализована модель, определяющая номер предложения, в котором текст, написанный человеком, сменяется текстом, сгенерированным искусственным интеллектом. Метод основан на анализе перплексии текста с использованием статистических словарей, построенных при помощи генеративных языковых моделей. В начале статьи выдвинута гипотеза, что перплексия напрямую зависит от типа написания текста (человеком или искусственным интеллектом) — в данной статье она подтверждена.

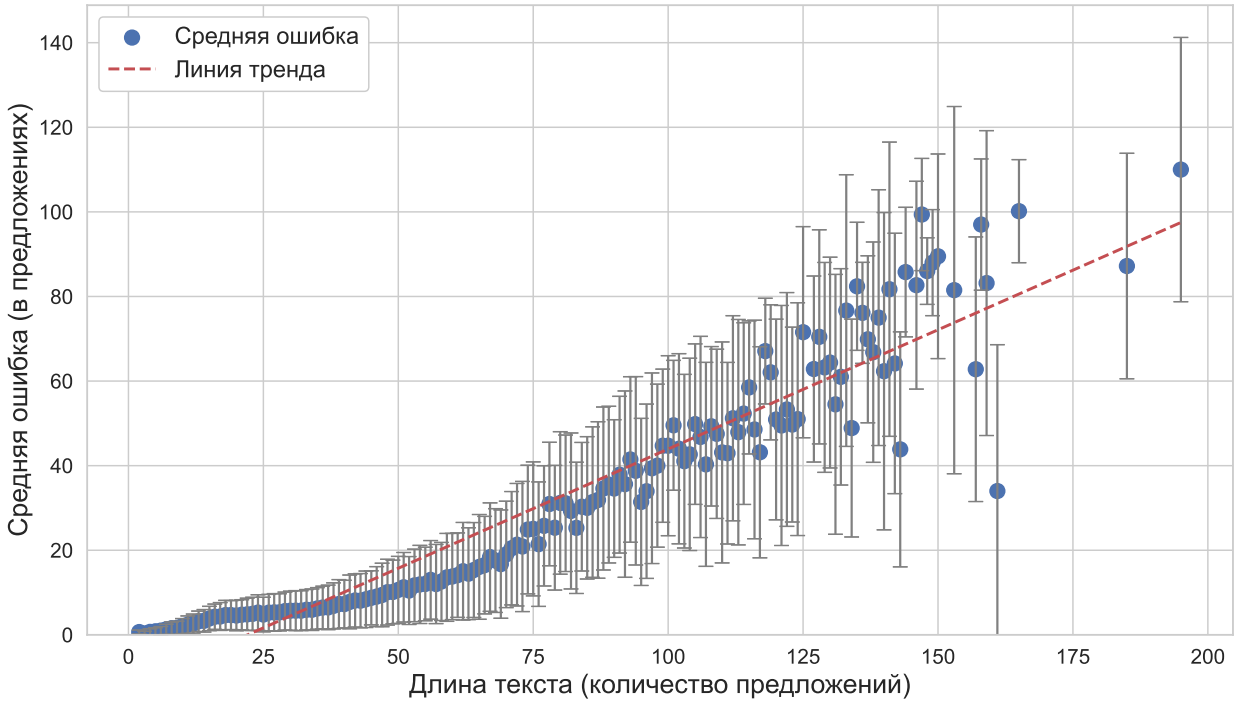

Рис. 5. Зависимость средней ошибки от длины текста.

Предложенный метод показывает стабильно высокие результаты на текстах длиной до 60-и предложений, что демонстрирует хорошую обобщающую способность построенной модели.

---

1. *G. Gritsai, I. Khabutdinov and A. Grabovoy.* Multihead span-based detector for AI-generated fragments in scientific papers, Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP), 2024.
2. *L. Dugan, D. Ippolito, A. Kirubarajan, S. Shi and C. Callison-Burch.* Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text, Proceedings of the AAAI Conference on Artificial Intelligence, 2023.
3. *G. Jawahar, M. Abdul-Mageed and L. Lakshmanan.* Automatic detection of machine generated text: A critical survey, CoRR (2020).
4. *P. P. Ray.* ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope, Internet of Things and Cyber-Physical Systems (2023).

5. *H. Touvron, T. Lavril, G. Izacard* et al.. LLaMA: Open and efficient foundation language models, arXiv:2302.13971 (2023).
6. *A. Jiang, A. Sablayrolles, A. Mensch* et al.. Mistral 7B, arXiv:2310.06825 (2023).
7. *A. Gray.* ChatGPT 'contamination': Estimating the prevalence of LLMs in the scholarly literature, arXiv:2403.16887 (2024).
8. *G. M. Gritsay, A. V. Grabovoy, A. S. Kildyakov* et al.. Artificially generated text fragments search in academic documents, Dokl. Math. **108**, S434–S442 (2023).
9. *K. Grashchenkov, A. Grabovoy and I. Khabutdinov.* A method of multilingual summarization for scientific documents, 2022 Ivannikov Ispras Open Conference (ISPRAS), 2022.
10. *G. Boeva, G. Gritsai and A. Grabovoy.* Team apteam at PAN: LLM adapters for various datasets, CLEF 2024: Conference and Labs of the Evaluation Forum, 2024.
11. *Yu. Chekhovich, A. Grabovoy and G. Gritsai.* Generative AI models with their full reveal, 2024 4th International Conference on Technology Enhanced Learning in Higher Education (TELE), 2024.
12. *E. Mitchell, Y. Lee, A. Khazatsky, C. Manning and C. Finn.* DetectGPT: Zero-shot machine-generated text detection using probability curvature, arXiv:2301.11305 (2023).
13. *A. Hans, A. Schwarzschild, V. Cherepanova and H. Kazem.* Spotting LLMs with binoculars: Zero-shot detection of machine-generated text, arXiv:2401.12070 (2024).
14. *X. Yang, W. Cheng, Y. Wu* et al.. DNA-GPT: Divergent N-gram analysis for training-free detection of GPT-generated text, arXiv:2305.17359 (2023).
15. *H. Wang, J. Li and Zh. Li.* AI-generated text detection and classification based on BERT deep learning algorithm, arXiv:2405.16422 (2024).
16. *Ch. Vasilatos, M. Alam, T. Rahwan* et al.. HowkGPT: Investigating the detection of ChatGPT-generated university student homework through context-aware perplexity analysis, arXiv:2305.18226 (2023).
17. *K. Heafield.* KenLM: Faster and smaller language model queries, Proceedings of the Sixth Workshop on Statistical Machine Translation (2011).
18. *H. Zhang and D. Chiang.* Kneser-Ney smoothing on expected counts, Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014).
19. *Yu. Wang, J. Mansurov, P. Ivanov* et al.. SemEval-2024 Task 8: Multidomain, multimodel and multilingual machine-generated text detection, arXiv:2404.14183 (2024).

20. *DAGPAP24: DETECTING AUTOMATICALLY GENERATED SCIENTIFIC PAPERS.* https://www.codabench.org/competitions/2431/
21. *Detecting AI Generated Content at COLING 2025.* https://genai-content-detection.gitlab.io/sharedtasks
22. *PAN at CLEF 2024: Voight-Kampff Generative AI Detection 2024.* https://pan.webis.de/clef24/pan24-web/generated-content-analysis.html
23. *PAN at CLEF 2025: Voight-Kampff Generative AI Detection 2025.* https://pan.webis.de/clef25/pan25-web/generated-content-analysis.html
24. *O. Bakhteev, A. Ogaltsov and P. Ostroukhov.* Fake news spreader detection using neural tweet aggregation?notebook for PAN at CLEF 2020, CLEF 2020 Labs and Workshops, Notebook Papers (2020).
25. *A. Uchendu, T. Le, K. Shu and D. Lee.* Authorship attribution for neural text generation, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020.
26. *G. M. Gritsaia, I. A. Khabutdinova and A. V. Grabovoya.* Stack More LLM's: Efficient Detection of Machine-Generated Texts via Perplexity Approximation, arXiv preprint (2023).
27. *P. Dmitrii, L. Mestetsky and A. Grabovoy.* N-Gram Perplexity-Based AI-Generated Text Detection, arXiv preprint (2024).
28. *A. Radford, J. Wu, R. Child, D. Luan, D. Amodei and I. Sutskever.* Language models are unsupervised multitask learners, OpenAI blog **1**(8), 9 (2019).
29. *M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov and L. Zettlemoyer.* BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 7871–7880 (2020).
30. *T. L. Scao, A. Fan, C. Akiki* et al.. BLOOM: A 176B-parameter open-access multilingual language model, arXiv preprint arXiv:2211.05100 (2022).
31. *S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin* et al.. OPT: Open Pre-trained Transformer language models, arXiv preprint arXiv:2205.01068 (2022).
32. *B. Guo, X. Zhang, Z. Wang* et al.. How close is ChatGPT to human experts? Comparison corpus, evaluation, and detection, arXiv:2301.07597 (2023).

33. *GPTZero.*

https://gptzero.me/

34. *Moxby.*

https://moxby.com/dashboard

35. *Copyleaks.*

https://copyleaks.com/ru/

BOUNDARY DETECTION TASK BETWEEN HUMAN AND MACHINE-GENERATED CONTENT

G. Pshenichnikov, A. Grabovoy

The purpose of this study is to develop an effective method for determining the sentence number with which the generated part of a hybrid text begins. The algorithm is based on sentence-level perplexity, calculated using complied probability dictionaries, which, along with other stylistic characteristics, is fed to the input of a binary classifier that determines whether the sentence is generated, and after analyzing the received labels, the text is divided into two parts. This study also confirms the hypothesis that perplexity depends on the type of text, whether it is generated or not, and when the type changes there occurs a sharp change in its value — a feature by which the sought transition boundary is found. The model achieves high accuracy — in texts containing no more than 12 sentences, the algorithm makes an error of no more than 1 sentence; in longer texts containing no more than 60 sentences, the algorithm makes an error of no more than 10 sentences. At the same time, the classifier used in the method achieves on average 80% accuracy and 80% recall in detecting a generated sentence.