Аннотация

В данной работе рассматривается задача детекции фрагментов машинно-сгенерированного в документе. Впервые задача детекции синтетического текста была поставлена как задача обработки естественного языка в 2019, вскоре после выхода первой открытой языковой модели GPT-2. В такой постановке задача является совокупностью двух различных задач, а именно, выделение фрагментов, различного авторства на основе смены стилистики текста. и его последующая классификация.

Цель данной работы состоит в разработке базовой модели нейронной сети для классификации искусственных фрагментов. Для реализации этого был проведен анализ существующих методов классификации текстов, изучены различные варианты архитектур нейронных сетей, пригодных для решения поставленной задачи.

В результате работы была разработана рабочая модель классификации, основанная на большой предобученной языковой модели типа трансформер. Для обучения и тестирования модели была использована специально подготовленная и размеченная выборка текстов, содержащих фрагменты, сгенерированные языковыми моделями. В работе подробно описывается процесс сбора и подготовки данных, процесс обучения моделей в различных вариантах нахождения фрагментов в документе, а также приводятся подробно результаты всех поставленных вычислительных экспериментов.

Содержание

1 Введение

Актуальность темы. В настоящий момент большие языковые модели стали повсеместно используемым инструментом для решения различных задач. Это стало особенно заметно после выхода чата-интерфейса ChatGPT [19] от компании OpenAI. Модель, на основе которой работал чат, значительно превосходила лучшие на тот открытые модели, и тексты, которая она генерировала, стали практически неотличимы от человеческого. Часто даже людям может быть сложно отличить человеческий текст от машинно-сгенерированного [4]. По этой причине ChatGPT и другие продвинутые языковые модели все чаще стали использоваться для решения повседневных, рабочих и учебных задач. Это, в свою очередь повлекло массовое использование сгенерированных текстов там, где ожидается текст, написанный человеком - например, в домашних заданиях и эссе [13], в текстах выпускных работ и научных статьях [11, 15]. Кроме того, часто языковые модели используются для генерации фейковых новостей [5, 14]. Большие языковые модели продолжают развиваться, их качество растет, поэтому должны развиваться и детекторы, с помощью которых можно было бы выделять инородные по стилю фрагменты. Задачу детекции машинно-сгенерированных текстов часто формулируют как задачу бинарной классификации автора текста. Однако при реальном использовании больших языковых моделей, часто ответы модели смешивают вместе с фрагментами, написанными человеком. Получается довольно сложная для решения задача, потому что:

- Сгенерированные фрагменты могут быть в произвольных позициях в документе.
- Сгенерированных фрагментов может быть сколько угодно много и они

могут быть произвольной длины.

- При генерации могут использоваться несколько больших языковых моделей.

Так как общая задача пока слишком сложная для решения, рассматриваются частные случаи этой задачи. Во-первых, можно ограничивать позиции документов, например разрешить смену автора только по предложениям или по параграфам. Во-вторых, можно разрешить только одну смену авторов, но в произвольном месте в документе. Также, не во всех работах рассматривается устойчивость детекторов к смене доменов или же к смене генерирующей модели и часто рассматривается только лишь один домен, чаще всего научный, новостной или домен студенческих эссе.

Цели работы.

1. Предложить метод детекции фрагментов различного авторства в документах в случае смены авторов по фиксированным позициям с помощью марковской линейной цепочки.

2. Предложить метод детекции фрагментов различного авторства в документах в случае единственной смены авторов — смены стиля.

Практическая значимость. Предложенные в работе методы могут использоваться для создания моделей детекции для поиска сгенерированного текста в реальных текстах, а также для измерения качества сгенерированного текста при обучении новых языковых моделей.

2 Обзор литературы

Существует довольно много работ, посвященных бинарной классификации автора текста [16, 24, 25], поэтому для решении задачи детекции гибридных текстов одним из возможных подходов является разбиение текста на фрагменты и решение подзадачи бинарной классификации для каждого фрагмента отдельно. Самым простым способом является подразбиение на предложения.

Пусть известно, что в документах может быть произвольное число смен авторов, но смена идет строго по предложениям. Так, в статье SeqXGPT [27] представлена модель, которая оценивает отдельно автора каждого предложения в документе. Одним из методов является решение задачи бинарной классификации для каждого предложения отдельно. Вторым методом, не предполагающим разбиение на предложения, является сопоставление меток “HUMAN” и “MACHINE” для каждого слова в документе. Метку же предложения предлагается определять большинством меток внутри предложения. Модель SeqXGPT основана как раз на втором способе и для сопоставления меток использует признаки, полученные из открытых моделей. Для документа $\overline{\mathbf{x}}$ и для токена $x_i \in \overline{\mathbf{x}}$ подсчитываются значения

$$ll_{\theta_n}(x_i) = \log p_{\theta_n}(x_i|x_{<i}),$$

где $\theta_1, \ldots, \theta_n$ - некоторые открытые языковые модели. Авторы предлагают воспринимать $\overline{ll_{\theta_n}(\overline{\mathbf{x}})}$ как понимание моделью θ_n семантики и синтаксиса текста $\overline{\mathbf{x}}$. Более продвинутые модели будут способны обрабатывать более сложные лингвистические структуры, поэтому и значение логарифма вероятности у них может быть больше. Помимо этого, полученные списки $ll_{\theta_n}(\overline{\mathbf{x}})$ довольно чувствительны к случайности сэмплирования, изменениям в обучающих данных, поэтому авторы предлагают рассматривать эти списки как волны.

Вдохновившись методами обработки сигналов, а именно свёртками, авторы предлагают использовать их для получения незашумленных значений. Применив сверточные слои, получившиеся скрытые переменные передаются на слой с вниманием, после которого наконец применяется стандартный линейный слой для классификации на классы. В качестве метрики авторы подсчитывают F1-меру относительно каждого предложения в тексте. SeqXGPT совсем немного обходит по метрикам базовое решение с RoBERTa [12], которая присваивает всем словам в документе метки “HUMAN” и “MACHINE” — 95.3 у SeqXGPT и 94.6 у RoBERTa. Дополнительно авторы показывают, что методы без обучения, такие как подсчет перплексии, использование DetectGPT [18] или Sniffer [10], не особенно хорошо работают на уровне предложений, в силу обычно короткой длины предложений.

Другой работой, в которой предварительно известно что смена авторов идет строго по предложениям, является работа, посвященная поиску сгенерированных фрагментов в учебных эссе [29]. Авторы рассматривают тексты, в которых есть от 1 до 3 смен авторов фрагментов — выбираются фрагменты человеческого текста и их позиции, и с помощью инструкции для ChatGPT генерируется оставшийся текст. Метод, предложенный в статье, основан на работе triplet-BERT [7] сетей — рассматриваются триплеты, состоящие из целевого предложения, из предложения с таким же авторством, что и целевое предложение, и из предложения с другим авторством. Сначала модель на основе трансформера дообучается так, чтобы косинусное расстояние между векторными представлениями предложений одного авторства было меньше чем между векторными представлениями предложений разного авторства. После получения представлений выбирается размер окна, внутри которого будут усредняться эти представления. Для каждого предложения отдельно усредняются представления внутри окна перед целевыми предложением, вме-

сте с векторными представлением целевого предложения, и представления в окне после целевого предложения. Для подсчета метрики берется топ-K индексов относительно расстояния между двумя усреднениями. Было показано, что дообучение модели BERT в данном случае является критичным и значительно помогает увеличить метрики, так как помогает обучить модель различать авторов, но тем не менее, даже в лучшем случае метрика F1-меры при использовании этого метода была равна 51.2.

В работах [6, 17] показано, что длина контекста имеет значение для бинарной классификации, и на коротких текстах, в том числе и на одном предложении, сложнее делать вывод о том, сгенерировано ли оно, чем делать такой же вывод, но для более длинных текстов. Поэтому в случае достаточно продвинутых моделей, например GPT-4 или LLaMA 3, которые генерируют тексты все более хорошего качества, может быть почти невозможно на уровне одного предложения определить его автора.

Другим вариантом подзадачи является постановка, в которой в документе сначала идет часть текста, написанная человеком, а затем часть текста, продолженная большой языковой моделью. Смена авторов может проходить по границе какого-то предложения, но не обязательно.

В работе [1] рассматриваются тексты из 10 предложений и требуется определить индекс, соответствующий первому предложению сгенерированного фрагмента. Для решения задачи были взяты тексты из датасета RoFT, описанном в главе 5.1. По сути, авторы решают задачу бинарной классификации на 10 предложениях. Они рассматривают несколько подходов, а именно классификация с помощью логистической регрессии на векторных представления отдельных предложений, полученных с помощью моделей RoBERTa и SRoBERTa [20], а также простое сравнение косинусных близостей пар соседних предложений. В качестве базового решения было предложено случайно

угадывать индекс предложения, брать преобладающий класс, а также было предложено базовое решение на основе подсчёта перплексии. Авторы провели два типа экспериментов — модели обучались либо на всем датасете, либо только на подмножестве, соответствующем какому-то определенному домену. Авторы сравнивали различные домены и различные генерирующий модели, и на разных комбинациях лучше сработали разные из трех методов. На большинстве доменов лидером стал подход на основе классификации векторных предложений от SRoBERTa, однако даже этот наилучший метод показал в среднем точность всего лишь 42%, что опять-таки подтверждает сложность задачи.

В работе [9] рассматривают несколько новых подходов. Первый подход основан на определении индекса смены с помощью некоторой BERT-модели, например с помощью RoBERTa. Передаются сразу все предложения и разделяются токеном [SEP]. Результат классификации записывается в [CLS] токен. Данный способ значительно превосходит по метрикам метод, в котором используется такая же BERT-модель, но которая отдельно классифицирует каждое предложение. Второй подход основан на получении значений перплексии из открытых моделей и классификации авторов с её помощью. В случае смены домена или стилей текстов именно этот подход показывает себя лучше всего и значительно обходит методы на основе дообучения. Наконец, авторы исследуют методы, в которых текст воспринимается как временной ряд с изменениями внутренней размерности текста [23]. Предлагается считать значение внутренней размерности текстовых эмбеддингов внутри некоторого скользящего окна. Далее полученный временной ряд классифицируется с помощью SVM. Авторы демонстрируют, что полученные распределения для человеческого и машинного текста отличаются и их возможно различаться, однако тем не менее, на данный момент этот метод уступает методам на основе дообучения и

перплексии. Однако этот метод устойчив к сменам доменов и моделей, поэтому его развитие может помочь делать более общие модели.

Данный подход хорошо подходит для случаев, когда домен на обучающих и тестовых данных одинаков или модель генерации не меняется, в ином случае результаты заметно ухудшаются. Это объясняется тем, что при дообучении модели склонны переобучаться на детекцию определенных признаков для конкретного домена или конкретной генерирующей модели.

Это подтверждает гипотезу, что все же детекция по предложениям довольно ограничена и в случае более сложных моделей будет проигрывать методам, которые работают с большими фрагментами текста.

Устойчивость моделей очень важна для качественной детекции, потому что на данный момент, почти любой детектор можно обмануть, подав как очень качественно сгенерированный текст от другой модели, так и слишком некачественный текст. Поэтому предлагается в дальнейшем использовать методы на основе перплексии, но при этом предобучать их на большом наборе синтетических данных от разнообразных моделей генерации.

3 Постановка задачи

Определим документ как конечную последовательность символов из заданного алфавита $\mathbf{W}$. Тогда пространство документов определено как:

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

Дан набор документов $\mathbf{D}$:

$$\mathbf{D} = \bigcup_{i=1}^{N} D^i, D^i \in \mathbb{D}.$$

Пусть известно, что для создания текстов в наборе $\mathbf{D}$ принимали участие k авторов, причем один из авторов обязательно человек. Определим множество авторов:

$$\mathcal{C} = \{0, \ldots, k-1\}.$$

При классификации будем помечать человеческий текст классом 0.

3.1 Детекция автора всего документа

Первой подзадачей является определение автора всего документа. Это задача классификации, в случае бинарной классификации необходимо определить, кто автор текста — человек или языковая модель. В случае многоклассовой задачи, автором текста может быть не только человек, но и какая-то модель из заранее известного набора языковых моделей. Формально, детектор определяется как

$$\phi : \mathbb{D} \to \mathcal{C}, \tag{1}$$

где $\mathcal{C} = \{0,1\}$ для бинарной детекции или $\mathcal{C} = \{0, \ldots, k-1\}$ для многоклассовой детекции и k языковых моделей-авторов. Так как это задача классифи-

кации, то метрикой качества здесь может быть точность в случае бинарной классификации и F1-мера в случае многоклассовой классификации.

3.2 Детекция фрагментов

Следующей подзадачей является нахождение в документе фрагментов другого авторства. Чаще всего подразумевается, что изначально текст написан человеком и в него добавлены фрагменты, написанные одной или несколькими языковыми моделями. Поэтому в таком случае, необходимо сначала разбить документ на фрагменты разного авторства, а после этого для каждого фрагмента необходимо определить его автора с помощью классификации, описанной в 3.1. Формально, для каждого документа $\mathbf{D} \in \mathbb{D}$ существует представление

$$\mathbb{T} = \left\{ \left[t_{s_j}, t_{f_j}, C_j \right]_{j=1}^{J} \quad | \quad t_{s_j} = t_{f_{j-1}}, \quad s_j \in \mathbb{N}_0, \quad f_j \in \mathbb{N}, \quad C_j \in \mathcal{C} \right\},$$

где J - количество фрагментов разного авторства, t_{s_j} и t_{f_j} - начало и конец j-ого фрагмента, внутри которого все токены одного авторства. В такой постановке предлагаемая модель детекции описывается композицией отображений

$$\phi : \mathbb{D} \to \mathbb{T} \qquad \phi : g \circ f, \tag{2}$$

где f — отображение для выделения текстовых фрагментов, а g - отображение для классификации получившихся фрагментов. В данном случае метрикой может быть процент пересечения предсказанных фрагментов с истинной разметкой фрагментов вместе с проверкой авторства.

3.3 Детекция смены стиля

Наконец, опишем частный случай задачи детекции фрагментов, когда известно, что смена авторов происходит единожды и причем смена автор-

ства строго с человеческого текста на машинно-сгенерированный. Т.е. для документа $d \in \mathbb{D}$ известно, что

$$\exists I \in \mathbf{N}_0 \quad g([t_0, t_I)) = 0, \quad g([t_{I+1}, t_{|d|})) = 1, \quad 0 \leq I < |d|,$$

В таком случае, задача сводится только к нахождению индекса единственного токена, где происходит смена автора с помощью отображения f. Метрикой качества детектора для такой задачи может служить значение ошибки предсказания положения индекса. Предлагается использовать метрику средней абсолютной ошибки, которая для набора документов считает среднее значение модуля разности между истинным положением индекса и предсказанным.

4 Методы детекции

В силу сложности задачи детекции машинно-сгенерированного текста — как бинарной детекции, так и выделения фрагментов — модель для решения этой задачи должна хорошо понимать естественный язык. В частности, модели должны уметь искать не только явные отличия в стилистике текстов, но и скрытые признаки искусственной природы текста. Например, модели генерации с детерминированным способом семплирования следующего слова в тексте довольно легко могут детектироваться в силу того, что можно подсчитать внутренние характеристики текста, такие как перплексия или логарифмическая вероятность и в случае совпадения смоделированного текста и поданного на детекцию текста, можно сделать вывод, что данный текст является машинно-сгенерированным. Однако, на данный момент, модели с детерминированной стратегией генерации почти не используются, поэтому чаще используют модели-нейронные сети на основе трансформерной [**?**] архитектуры. Слой внимания в трансформерах позволяет учитывать контекст, что особенно важно для детекции фрагментов внутри документа, ведь необходимо сравниваться с другими фрагментами документами.

Кроме того, неоспоримым плюсом построения модели на основе трансформеров является наличие большого числа уже предобученных моделей, которые можно найти в открытом доступе. Взяв предобученную модель,которая уже обладает хорошим уровнем понимания естественного языка, можно уже дообучить ее под конкретную задачу — например, на задачу бинарной классификации текста. Помимо этого, с точки зрения сложности реализации и затрат вычислительных ресурсов, дообучение существующей модели под конкретную задачу гораздо выгоднее, чем обучение модели с нуля. В данной главе векторные представления текстовых единиц, таких как предложения и параграфы,

получаются с помощью токенизации с помощью предобученного энкодера на основе модели BERT [2]. BERT является одной из классических трансформерных моделей и на данный момент использование различных предобученных моделей на его основе является стандартным подходом токенизации текстов.

4.1 Модель для бинарной детекции

Так как в данной подзадаче требуется определить автора всего документа, то для классификации предлагается векторные представления документов передавать в линейный слой нейронной сети и после получения логитов выбирать автора документа по токену [CLS], который является специальным токеном для всех BERT-энкодеров, предназначенным для классификации.

Формально, пусть для $d \in \mathbb{D}$ с помощью BERT или другого энкодера были получены векторные представления $\mathbf{h} = (\mathbf{h}_{\text{CLS}}, \mathbf{h}_1, ..., \mathbf{h}_n)$. Тогда автором документа будет:

$$\hat{y} = \arg\max_{c}(\text{softmax}(\mathbf{W} \cdot \mathbf{h}_{\text{CLS}} + \mathbf{b})),$$

где $\mathbf{W}, \mathbf{b}$ - обучаемые параметры линейного слоя .

4.2 Модель для детекции фрагментов

Если известно по каким фрагментам проходит граница, то задача сводится к задаче бинарной классификации фрагментов и агрегированию полученных метрик бинарной классификации. Однако, в данном случае если каждый фрагмент оценивается отдельно, то никак не учитывается контекст фрагмента, например предыдущий параграф или предложение, а это может быть полезно при детекции автора текущего фрагмента. Для того чтобы учитывать контекст, предлагается использовать модель с марковским случайным полем (от англ.

Conditional Random Fields), а точнее с марковской линейной цепочкой.

Пусть для документа $d \in \mathbb{D}$ известны его параграфы $\mathcal{P} = (p_1, \ldots, p_n)$ и $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ — векторные представления соответствующих параграфов. Необходимо построить вероятностную модель

$$p(y_1, \ldots, y_n | \mathbf{x}_1, \ldots, \mathbf{x}_n) = p(\mathbf{y}|\mathbf{x})$$

где $\mathbf{x}_i$ — векторное представление параграфа p_i, y_i — автор параграфа p_i.

Ключевой идеей моделей с марковскими случайными полями является введение вектора признаков

$$\Phi(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d$$

Функция Φ отображает последовательность векторных представлений и последовательность тегов-авторов в некоторый вектор признаков в пространстве $\mathbb{R}^d$, при этом каким-то образом учитывая зависимости между $\mathbf{x}_i$ и y_i и между соседними тегами y_i и y_{i-1}. Тогда вероятность получить последовательность тегов для данного документа выражается как:

$$p(\mathbf{y}|\mathbf{x}) = \frac{\exp \Phi(\mathbf{x},\mathbf{y})}{\sum\limits_{\mathbf{y}' \in \mathcal{Y}^n} \exp(\Phi(\mathbf{x},\mathbf{y}'))}, \tag{3}$$

где $\mathcal{Y}^n$ – множество всех возможных последовательностей тегов-авторов длины n — в случае бинарной задачи это, по сути, все последовательности из 0 и 1 длины n. В общем случае, можно записать следующую зависимость:

$$p(\mathbf{y}|\mathbf{x}) \propto \prod_{i=1}^{n} \exp(\phi_i(y_{i-n}, \ldots, y_i, \mathbf{x}, i),$$

где функции ϕ_i отвечают за извлечение полезных для определения авторства признаков и определены как

$$\Phi = \sum_{i=1}^{n} \log \phi_i(y_{i-k}, \ldots, y_i, \mathbf{x}, i),$$

где k является размером контекстного окна, которое влияет на текущее предсказание. Для задачи детекции достаточно будет взять $k = 1$ и зависеть только от предыдущего параграфа.

Перейдем к рассмотрению вида непосредственно функций ϕ_i. Предлагается использовать две функции, первая из которых, $\phi_{\texttt{EMIT}}$, соотносит вероятность сопоставления тега y_i с векторным представлением $\mathbf{x}_i$. Данная функция называется "выхлопом" (от англ. emissions), так как эти значения будут передаваться слою с марковской линейной цепочкой от какой-то другой модели, например из энкодера, LSTM-слоя или полносвязного слоя. На рис. 1 показана схема использования CRF-слоя вместе с энкодером на основе BERT. Вторая функция, $\phi_{\texttt{TRANS}}$, устанавливает вероятности появления меток y_i и y_{i-1} в качестве соседей. Данная функция обычной называется "переходом" (от англ. transitions). Таким образом, функция Φ определена как:

$$\Phi(\mathbf{x}, \mathbf{y}) = \log \phi_{\texttt{EMIT}}(y_1, \mathbf{x}_1) + \sum_{i=2}^{n} (\log \phi_{\texttt{EMIT}}(y_i, \mathbf{x}_i) + \log \phi_{\texttt{TRANS}}(y_{i-1}, y_i)). \quad (4)$$

Задача обучения ставится как задача минимизации негативного логарифмического правдоподобия (Negative Log-Likelihood):

$$\mathcal{L}(\hat{\mathcal{Y}}, \mathcal{D}) = -\sum_{i=1}^{|\mathcal{D}|} \log(p(\mathbf{y}^i|\mathbf{x}^i)) = -\sum_{i=1}^{|\mathcal{D}|} \log \Big[\frac{\exp \Phi(\mathbf{x}^i, \mathbf{y}^i)}{\sum\limits_{\mathbf{y}' \in \mathcal{Y}^n} \exp(\Phi(\mathbf{x}^i, \mathbf{y}'))} \Big] =$$

$$= \sum_{i=1}^{|\mathcal{D}|} \Big(\log \Big[\sum_{\mathbf{y}' \in \mathcal{Y}^n} \exp(\Phi(\mathbf{x}^i, \mathbf{y}')) \Big] - \Phi(\mathbf{x}^i, \mathbf{y}^i) \Big), \quad (5)$$

где $\mathcal{D}, \hat{\mathcal{Y}}$, - обучающая выборка. $\mathbf{x}^i \in \mathcal{D}, \mathbf{y}^i \in \hat{\mathcal{Y}}$,. Основная вычислительная сложность лежит в вычислении первого выражения в слагаемых в выражении 5, так как необходимо вычислить переходы для всех возможных последователь-

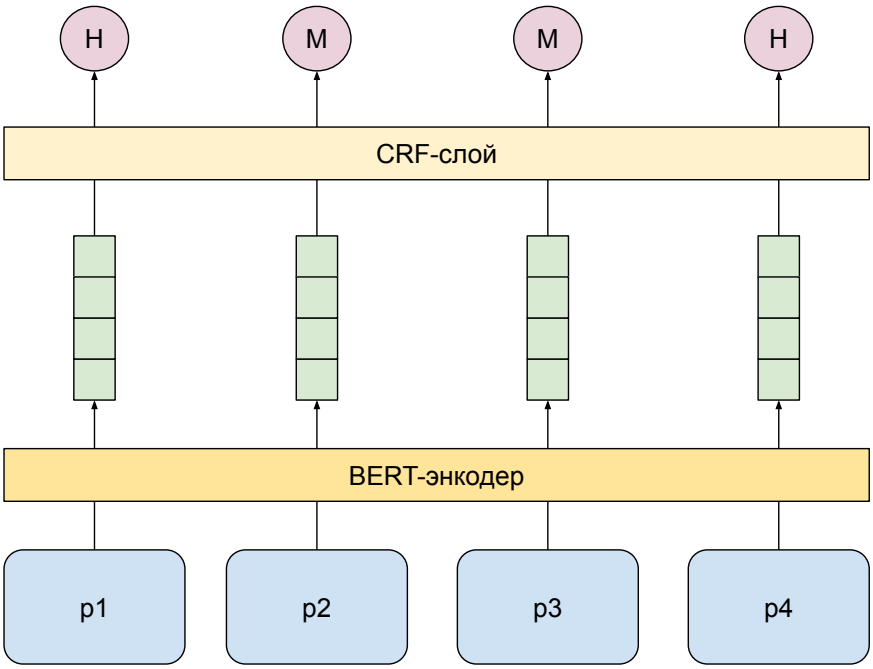


Рис. 1: Схема модели с марковской линейной цепочкой. После применения модели каждому параграфу документа присваивается метка “Н” для параграфа, написанного человеком и метка “М” для параграфа, сгенерированного большой языковой моделью

ностей меток-авторов, что в случае очень длинных последовательностей или последовательностей в большим количеством возможных авторов невозможно. Однако, используя динамическое программирование и алгоритм Витерби [8] можно достаточно эффективно вычислить знаменатель.

Обозначим матрицу переходов, состоящую из значений функции $\phi_{\texttt{TRANS}}$, как T:

$$T_{i,j} := \phi_{\texttt{TRANS}}(y_i, y_j). \tag{6}$$

Введем функцию π_i^j, которая будет вычисляться как логарифм всех значений последовательностей от 1-ого параграфа и до $i+1$-ого параграфа, так, что метка последнего параграфа будет равен $j \in \mathcal{C}$, $1 \leq i \leq |\mathbf{x}|$:

$$\pi_i(\mathbf{x}) := \log \sum_{\mathbf{y} \in \mathcal{Y}^i} \exp(\Phi(\mathbf{x},\mathbf{y})), \tag{7}$$

$$\pi_i^j(\mathbf{x}) := \log \sum_{\substack{\mathbf{y}\in\mathcal{Y}^i \\ y_i=j}} \exp(\Phi(\mathbf{x},\mathbf{y})), \tag{8}$$

База рекурсии:

$$\pi_0(\mathbf{x}) = T_{0,\cdot}(\mathbf{x}) + \phi_{\texttt{EMIT}}(y_0, x_0). \tag{9}$$

Докажем, что мы можем определять π_i^j рекурсивно.

Утверждение 1. *$\pi_i^j(\mathbf{x})$ представима в виде $\psi_{ij}(\mathbf{x})$, где*
$\psi_{ij}(\mathbf{x}) = \log \sum_{t=1}^{|\mathcal{C}|} \exp\Big[\pi_{i-1}^j(\mathbf{x}) + \log \phi_{\textit{EMIT}}(y_i, \mathbf{x}_i) + \log \phi_{\textit{TRANS}}(t, j)\Big]$.

Доказательство. Докажем, что рекурсия корректна. Пусть $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ - векторные представления некоторого документа из $\mathbb{D}$. По определению $\Phi(\mathbf{x},\mathbf{y}')$

$$\pi_{i-1}^j(\mathbf{x}) = \log \sum_{\substack{\mathbf{y}\in\mathcal{Y}^{i-1} \\ y_{i-1}=j}} \exp\Big[\sum_{k=1}^{i-1}\Big(\log \phi_{\texttt{EMIT}}(y_k, \mathbf{x}_k) + \log \phi_{\texttt{TRANS}}(y_{k-1}, y_k)\Big)\Big] \tag{10}$$

$$\pi_i^j(\mathbf{x}) = \log \sum_{\substack{\mathbf{y}\in\mathcal{Y}^{i} \\ y_{i}=j}} \exp\Big[\sum_{k=1}^{i}\Big(\log \phi_{\texttt{EMIT}}(y_k, \mathbf{x}_k) + \log \phi_{\texttt{TRANS}}(y_{k-1}, y_k)\Big)\Big] \tag{11}$$

Распишем правую часть равенства 11, сделав "шаг назад" и перебрав все возможные значения меток, которые могут стоять на $i-1$ позиции:

$$\begin{aligned}\pi_i^j(\mathbf{x}) = \log \sum_{t=0}^{|\mathcal{C}|} \sum_{\substack{\mathbf{y}\in\mathcal{Y}^{i} \\ y_{i-1}=t}} \exp\Big[&\log \phi_{\texttt{EMIT}}(y_i, \mathbf{x}_i) + \log \phi_{\texttt{TRANS}}(t, j) + \\ &+ \sum_{k=1}^{i-1}\Big(\log \phi_{\texttt{EMIT}}(y_k, \mathbf{x}_k) + \log \phi_{\texttt{TRANS}}(y_{k-1}, y_k)\Big)\Big] = \\ = \log \sum_{t=0}^{|\mathcal{C}|} \sum_{\substack{\mathbf{y}\in\mathcal{Y}^{i} \\ y_{i-1}=t}} \exp\Big[\sum_{k=1}^{i-1}&\Big(\log \phi_{\texttt{EMIT}}(y_k, \mathbf{x}_k) + \log \phi_{\texttt{TRANS}}(y_{k-1}, y_k)\Big)\Big] \cdot \\ &\cdot \phi_{\texttt{EMIT}}(y_i, \mathbf{x}_i) \cdot \phi_{\texttt{TRANS}}(t, j)\end{aligned}$$

С учетом равенства 10

$$\pi_i^j(\mathbf{x}) = \log \sum_{t=0}^{|\mathcal{C}|} \Big[\exp(\pi_{i-1}^t(\mathbf{x})) \cdot \phi_{\texttt{EMIT}}(y_i, \mathbf{x}_i) \cdot \phi_{\texttt{TRANS}}(t, j) \Big] =$$
$$= \log \sum_{t=0}^{|\mathcal{C}|} \exp \Big[\pi_{i-1}^t(\mathbf{x}) + \log \phi_{\texttt{EMIT}}(y_i, \mathbf{x}_i) + \log \phi_{\texttt{TRANS}}(t, j) \Big]$$

□

Таким образом, умея быстро подсчитывать функцию π_i^j, мы можем и быстро считать функцию потерь 5.

Чтобы восстанавливать последовательность меток для некоторого документа с помощью марковской цепочки, формально нужно найти такую последовательность меток, которая является наиболее вероятной для данного документа $\mathbf{x}$:

$$\arg\max_{\mathbf{y} \in \mathcal{Y}^n} p(\mathbf{y}'|\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}^n} \frac{\exp(\mathbf{\Phi}(\mathbf{x}, \mathbf{y}))}{\sum\limits_{\mathbf{y}' \in \mathcal{Y}^n} \exp(\mathbf{\Phi}(\mathbf{x}, \mathbf{y}')))} = \arg\max_{\mathbf{y} \in \mathcal{Y}^n} \mathbf{\Phi}(\mathbf{x}, \mathbf{y}) =$$
$$= \arg\max_{\mathbf{y} \in \mathcal{Y}^n} \sum_{i=2}^{n} (\log \phi_{\texttt{EMIT}}(y_i, \mathbf{x}_i) + \log \phi_{\texttt{TRANS}}(y_{i-1}, y_i))$$

Данные значения можно считать точно также с помощью динамического программирования, и функции π_i^j, которая хранит максимальное суммарное значение $\Phi(\mathbf{x}, \mathbf{y})$ для всех последовательностей длины $i+1$, последняя метка которых соответствует j. Сохраняя на каждом шаге индексы, на которых достигается максимум, мы можем восстановить последовательность тегов.

4.3 Модель для детекции единственной смены авторов

Для данной подзадачи достаточно с помощью модели f найти позицию смены авторства. Назовем эту позицию сменой стиля. Параметризуем f дли-

ной скользящего окна ℓ. Функция f будет для каждого токена в документе оценивать вероятность его позиции быть позицией смены стиля:

$$f_\ell : \mathbb{D} \times \mathbb{N}_0 \to [0,1] \quad f_\ell(d,i) = p(t_i | t_{i-\ell}, \ldots, t_{i-1}, t_{i+1}, t_{i+\ell}), \tag{12}$$

$$I_d := \arg \max_{0 \le i \le |d|} f_\ell(d,i) \tag{13}$$

5 Вычислительные эксперименты

5.1 Обзор существующих датасетов

Существует много датасетов, содержащих синтетические тексты, большинство из которых предназначено для решения задачи бинарной классификации. Для решения задачи бинарной классификации использовался датасет с текстами на русском языке от различных моделей RuATD [21].

Датасетов, содержащих именно гибридные тексты не так много в силу сложности сбора таких датасетов. Большинство существующих датасетов предназначено для решения задачи нахождения границы по предложениям. Так, первым датасетом стал Real or Fake Text (RoFT) [3] — для его создания авторы дополняли тексты с помощью моделей GPT-2. В текстах датасета сначала идет часть, написанная человеком, а далее — часть, сгенерированная языковыми моделями с помщощью человеческого префикса. Ещё один датасет, RoFT-chatgpt [9], дополняет RoFT генерациями от ChatGPT в таком же формате. Наконец, в датасете SeqXGPT [27] для генерации использовались открытые модели, но формат остался таким же. Датасетов, в которых бы смена авторов была по параграфам, в открытом доступе найдено не было, поэтому для проведения экспериментов со сменой стиля по определенным позициям было решено сгенерировать свой датасет в котором бы смена авторов шла ровно по параграфам. Создание и характеристики этого датасета описаны в главе 5.2.2.

Наконец, недавно было проведено несколько соревнований, в которых участникам предлагалось решить различные задачи поиска сгенерированного текста. Одним из таким соревнованием является SemEval2024 Task 8 Subtask C [28], в котором предлагалось найти границу между началом, написанным

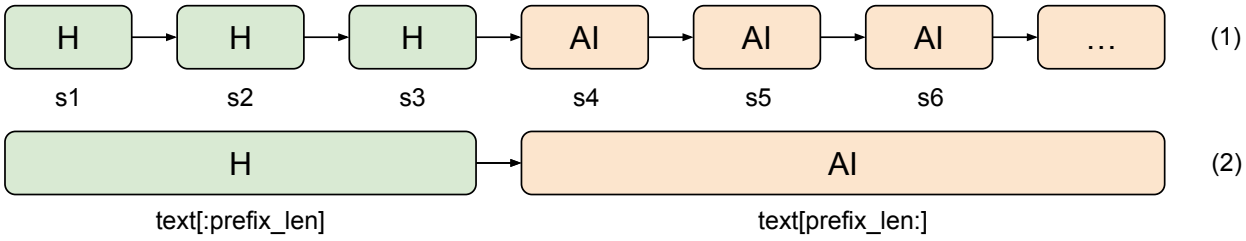


Рис. 2: Схемы авторства текстов в различных датасетах. Тег “H” соответствует человеческому фрагменту, “AI” — фрагменту, сгенерированному некоторый языковой моделью. (1) соответствует датасетам RoFT, RoFT-chatgpt и SeqXGPT, (2) - датасету SemEval2024 Task 8

человеком, и продолжением, сгенерированным с помощью языковой модели. В отличии от предыдущих датасетов, граница могла проходить в любом месте текста и по сути необходимо было решить задачу смены стиля. Датасет для соревнования был составлен с помощью моделей ChatGPT и LLaMA 2 [22]. При этом в данном датасете встречается много текстов с некачественной генерацией [26] — например, в начале генерации модель начинает повторять человеческий префикс и это сильно упрощает задачу для детектора.

Название датасета	Модели для генерации	Тип смены
RoFT	GPT-2	по предложениям
RoFT-chatgpt	GPT-2, ChatGPT	по предложениям
SeqXGPT	GPT-2, GPT-Neo, GPT-J, LLaMA	по предложениям
SemEval2024	ChatGPT, LLaMA 2	смена стиля

Таблица 1: Сравнение различных датасетов для детекции фрагментов

В данной работе для решения подзадач детекции фрагментов был использован датасет SemEval2024 Task 8 Subtask C.

5.2 Описание экспериментов

5.2.1 Бинарная классификация

Для решения задачи бинарной классификации сравнивалось несколько дообученных моделей на основе архитектуры BERT, некоторые из которых были мультилингвальными — BERT, mDeBERTa-V3, XLM-RoBERTa и ruBERT. Дополнительно было дообучены модели RoBERTa и DeBERTa. Метрикой для сравнения была выбрана точность классификации.

Метод	**Точность**
TF-IDF	0.64223
BERT-base-multilingual	0.73430
RoBERTa-base	0.63847
DeBERTa-v3-base	0.72661
mDeBERTa-v3-base	0.76662
XLM-RoBERTa-base	0.72661
XLM-RoBERTa-large	0.76777
ruBERT	0.77288

Таблица 2: Точность бинарной классификации различных подходов. TF-IDF был выбран в качестве бейзлайнового решения

Так как датасет, на котором дообучались и оценивались модели был русскоязычным, то у моделей, которые были изначальны обучены на мультилингвальных или русскоязычных текстов, получилось более точно классифицировать тексты из тестового датасета, чем у таких же, но обученных только на английских текстах. Наилучшее качество было получено после обучения модели ruBERT — 77%.

5.2.2 Детекция фрагментов

Для бинарной сегментации мы сгенерировали набор данных из 10000 документов следующим образом: сначала мы взяли открытый датасет со статьями с сайта Medium.com. Обрезав статьи до длины 4000 токенов, мы случайным образом выбрали до 3 абзацев, которые будут заменены на сгенерированные машиной фрагменты. Для машинной генерации мы взяли открытую модель LLaMA, так как на момент генерации это была наилучшая по качеству открытая модель. Для каждого выбранного абзаца мы давали модели предыдущий абзац в качестве промпта. После генерации искусственного абзаца мы обрезали его, чтобы его длина не превышала 700 токенов, и помещали его на место оригинального абзаца в документе. Наш набор данных состоит из 4000 документов с 3 замененными абзацами, 2500 документов с 4 замененными абзацами и 3500 документов с 2 замененными абзацами.

Для классификации параграфов была использована модель RoBERTa - параграфы классифицировались отдельно и их метки не зависели от меток соседних параграфов. После этого были проведены эксперименты с добавлением CRF-слоя, описанного в главе 4.2. Использование CRF позволило поднять среднюю точность классификации с 89% до 93%. При использовании таких же моделей, но для разделения по предложениям, результат получился значительно хуже — всего 74% и 75% точности соотвественно.

Дополнительно мы сравнили, как хорошо раздеялются фрагменты разного авторства. Предполагается, что взяв векторные представления двух фрагментов из одного документа разного авторства и посчитав косинусное расстояние между ними, мы получим отрицательное значение, близкое к 1, а в случае фрагментов из одного документа одного автора, косинусное расстояние должно быть положительно и близко к 1. На рис. 3 представлены

результаты подсчета p косинусных расстояний для векторный представлений соседних параграфов и соседних предложений. Видно, что параграфы хорошо разделяются, а вот предложения не так хорошо, что подтверждает вывод о том, что слишком короткие тексты, например предложения, не подходят для детекции машинно-сгенерированного текста [6, 17], так детектору не хватает информации для уверенной классификации.

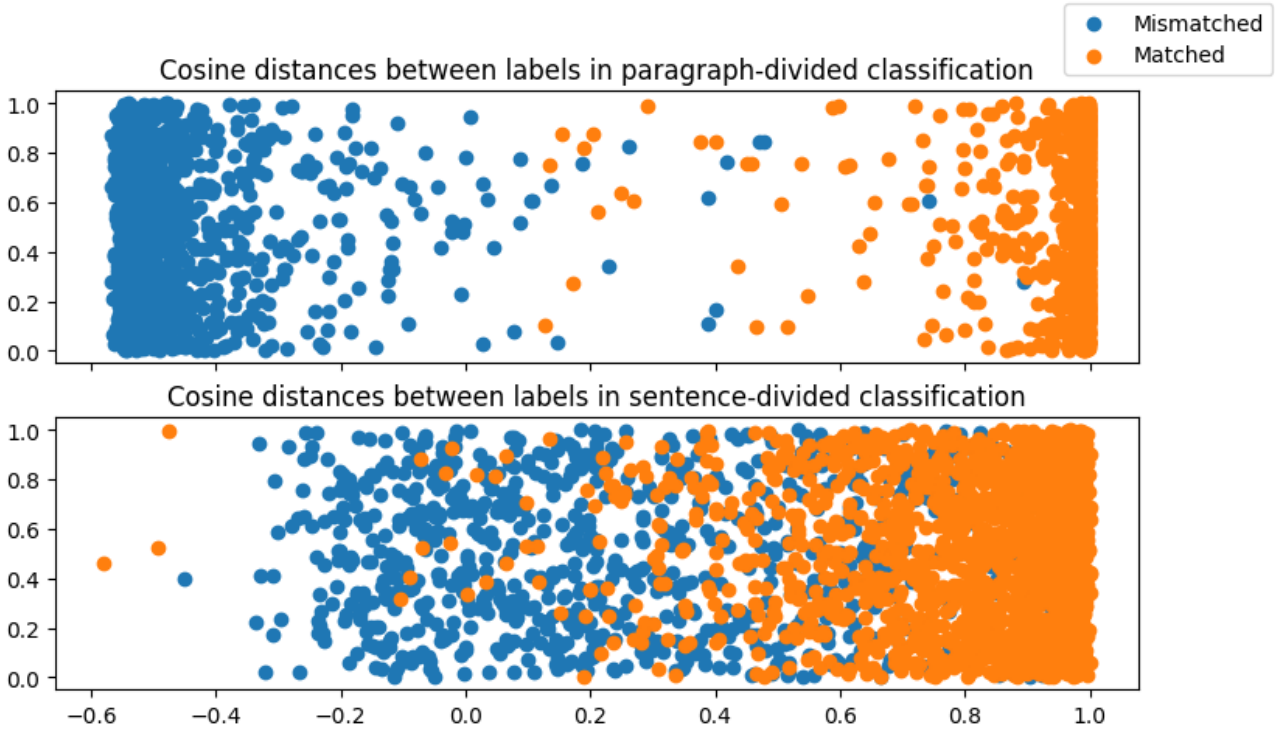


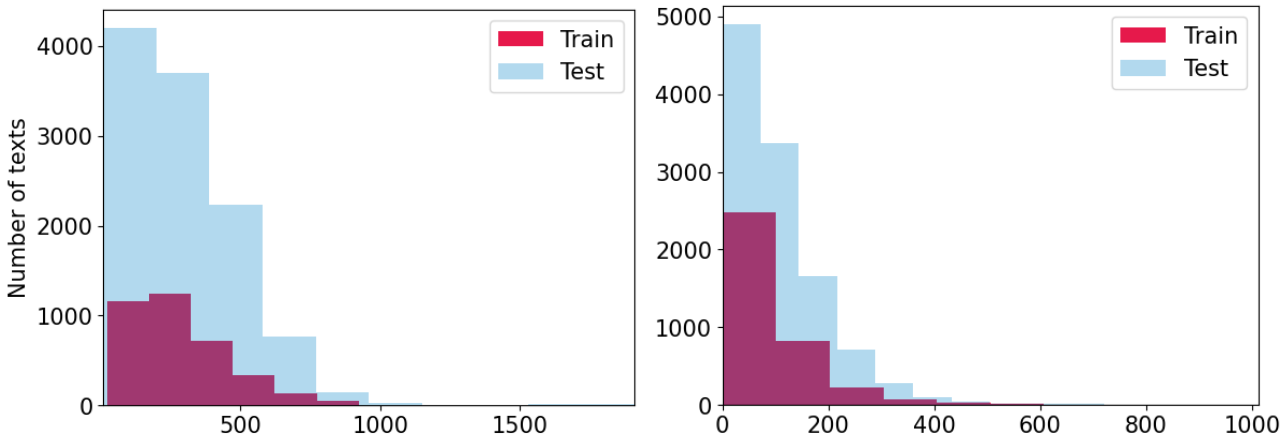
Рис. 3: Расстояния между векторными представлениями фрагментов разного авторства в случае, если а) фрагментами являются параграфы б) фрагментами являются предложения

5.2.3 Детекция смены стиля

Датасет SemEval2024 Task 8 Subtask C оказался достаточно маленьким, поэтому для получения обучающей выборки достаточно большого размера и для внесения большего разнообразия в положении индекса смены авторов

предварительно была сделана аугментация данных. До аугментации данных в обучающей выборке было 3648 текстов, после аугментации получилось 8059 текстов.

На рис. 4 показаны две основные статистики текстов в исходных датасетах, которые были полезны для решения — длины текстов и положение индекса смены авторов. Почти все тексты в обучающей и тестовой выборке были достаточно короткими — до 1000 токенов. Видно, что в большинстве текстов смена авторов происходит в окне с 0 по 400 токен, поэтому для токенизатора даде окна в 512 токенов было достаточно и на таком размере окна результаты получились даже лучше, чем на более длинном окне.



(a) Распределение длин текстов

(b) Положение индекса смены стиля

Рис. 4: Статистики текстов в датасете после токенизации

Все эксперименты были проведены на двух наборах данных - на исходных данных от организаторов и на аугментированных данных. Сравнивались три модели: Longformer, RoBERTa и DeBERTaV3. Организаторы соревнования предлагали использовать Longformer-base в качестве базового решения. Результаты экспериментов представлены в таблице 3.

Из данных экспериментов можно сделать следующий вывод: даже простая аугментация данных позволяет довольно сильно улучшить результаты работы

Модель	**Исходный датасет**	**Новый датасет**
RoBERTa-base	31.56	30.71
RoBERTa-large	25.25	20.66
longformer-base	23.16	22.94
longformer-large	22.97	20.33
DeBERTaV3-base	16.12	13.98
DeBERTaV3-large	15.16	**13.38**
Top 1 соревнования	15.68	-

Таблица 3: Метрика MAE на исходных и новых (аугментированных) данных. Дополнительно приведено лучшее решение, получившее первое место по результатам соревнования

метода. Кроме того, что данный метод значительно превосходит по метрике не только решение организаторов с longformer-base с результатом 21.15 MAE, но и лучшее решение с соревнования, равное 15.68 MAE.

6 Заключение

В данной работе были рассмотрены различные задачи, связанные с детекцией машинно-сгенерированных текстов. Была рассмотрена задача бинарной детекции, когда необходимо определить автора всего документа. В качестве вычислительного эксперимента сравнивалось результаты дообучения различных энкодеров на датасете с русскими текстами.

Другой задачей является детекция смены авторов по фиксированным позициями, например по параграфами или предложениям. Предложено два подхода, первый из которых рассматривает каждый фрагмент как отдельный текст и сводит задачу к предыдущей. Второй подход предлагает использовать марковские линейные цепочки для учета контекста фрагментов. Учет контекста помогает немного улучшить результаты детекции. Кроме того, было показано, что детекция по параграфам позволяет детектору быть более уверенным в своих предсказания, нежели при детекции по предложениям, что подтверждает выводы в других работах о том, что слишком маленькие тексты могут ухудшать качество работы детектора.

Наконец, последней подзадачей является задача поиска единственной смены стиля, когда известно, что сначала идет человеческий текст, а потом идет машинный текст, но при этом смена авторов может быть в произвольной позиции текста. Для данной подзадачи предлагается опять использовать предобученные модели энкодеров. Методы, предложенные для решения этой задачи, на данный момент показывают наилучшие результаты по итогам соревнования SemEval2024 Task 8, посвященному решению этой задачи.

Список литературы

[1] Cutler, J. W., L. Dugan, S. Havaldar, and A. Stein (2021). Automatic detection of hybrid human-machine text boundaries.

[2] Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

[3] Dugan, L., D. Ippolito, A. Kirubarajan, and C. Callison-Burch (2020, October). RoFT: A tool for evaluating human detection of machine-generated text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, pp. 189–196. Association for Computational Linguistics.

[4] Dugan, L., D. Ippolito, A. Kirubarajan, S. Shi, and C. Callison-Burch (2022). Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *AAAI Conference on Artificial Intelligence*.

[5] Fagni, T., F. Falchi, M. Gambini, A. Martella, and M. Tesconi (2020). Tweepfake: About detecting deepfake tweets. *PLoS ONE 16*.

[6] Gritsay, G., A. Grabovoy, and Y. Chekhovich (2022). Automatic detection of machine generated texts: Need more tokens. In *2022 Ivannikov Memorial Workshop (IVMEM)*, pp. 20–26.

[7] Hoffer, E. and N. Ailon (2014). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*.

[8] Jurafsky, D. and J. H. Martin (2009). *Speech and language processing* (2. ed.,

[Pearson International Edition] ed.). Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International.

[9] Kushnareva, L., T. Gaintseva, G. Magai, S. Barannikov, D. Abulkhanov, K. Kuznetsov, E. Tulchinskii, I. Piontkovskaya, and S. Nikolenko (2024). Ai-generated text boundary detection with roft.

[10] Li, L., P. Wang, K. Ren, T. Sun, and X. Qiu (2023). Origin tracing and detecting of LLMs.

[11] Liang, W., Y. Zhang, Z. Wu, H. Lepp, W. Ji, X. Zhao, H. Cao, S. Liu, S. He, Z. Huang, D. Yang, C. Potts, C. D. Manning, and J. Y. Zou (2024). Mapping the increasing use of llms in scientific papers.

[12] Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). RoBERTa: A robustly optimized bert pretraining approach. *ArXiv abs/1907.11692*.

[13] Liu, Y., Z. Zhang, W. Zhang, S. Yue, X. Zhao, X. Cheng, Y. Zhang, and H. Hu (2023). ArguGPT: evaluating, understanding and identifying argumentative essays generated by gpt models.

[14] Loth, A., M. Kappes, and M.-O. Pahl (2024). Blessing or curse? a survey on the impact of generative ai on fake news.

[15] Ma, Y., J. Liu, F. Yi, Q. Cheng, Y. Huang, W. Lu, and X. Liu (2023). AI vs. human – differentiation analysis of scientific content generation.

[16] Macko, D., R. Moro, A. Uchendu, J. Lucas, M. Yamashita, M. Pikuliak, I. Srba, T. Le, D. Lee, J. Simko, and M. Bielikova (2023, December). MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In H. Bouamor,

J. Pino, and K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 9960–9987. Association for Computational Linguistics.

[17] Mireshghallah, N., J. Mattern, S. Gao, R. Shokri, and T. Berg-Kirkpatrick (2024, March). Smaller language models are better zero-shot machine-generated text detectors. In Y. Graham and M. Purver (Eds.), *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, St. Julian's, Malta, pp. 278–293. Association for Computational Linguistics.

[18] Mitchell, E., Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn (2023). DetectGPT: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23.

[19] OpenAI (2023). Gpt-4 technical report.

[20] Reimers, N. and I. Gurevych (2019). Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

[21] Shamardina, T., V. Mikhailov, D. Chernianskii, A. Fenogenova, M. Saidov, A. Valeeva, T. Shavrina, I. Smurov, E. Tutubalina, and E. Artemova (2022). Findings of the the ruatd shared task 2022 on artificial text detection in russian.

[22] Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan,

M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom (2023). Llama 2: Open foundation and fine-tuned chat models.

[23] Tulchinskii, E., K. Kuznetsov, L. Kushnareva, D. Cherniavskii, S. Barannikov, I. Piontkovskaya, S. I. Nikolenko, and E. Burnaev (2023). Intrinsic dimension estimation for robust detection of ai-generated texts. *ArXiv abs/2306.04723*.

[24] Uchendu, A., T. Le, K. Shu, and D. Lee (2020, November). Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8384–8395. Association for Computational Linguistics.

[25] Uchendu, A., Z. Ma, T. Le, R. Zhang, and D. Lee (2021, November). TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic, pp. 2001–2016. Association for Computational Linguistics.

[26] Voznyuk, A. and V. Konovalov (2024). DeepPavlov at SemEval-2024 Task 8: Leveraging Transfer Learning for Detecting Boundaries of Machine-Generated Texts.

[27] Wang, P., L. Li, K. Ren, B. Jiang, D. Zhang, and X. Qiu (2023, December). SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing, Singapore, pp. 1144–1156. Association for Computational Linguistics.

[28] Wang, Y., J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. M. Afzal, T. Mahmoud, G. Puccetti, T. Arnold, A. F. Aji, N. Habash, I. Gurevych, and P. Nakov (2024, June). Semeval-2024 task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico, Mexico.

[29] Zeng, Z., L. Sha, Y. Li, K. Yang, D. Gašević, and G. Chen (2023). Towards automatic boundary detection for human-AI collaborative hybrid essay in education.