ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет о командном программном проекте на тему:

Анализ и классификация текстов с целью выявления авторства искусственного интеллекта: исследование лексических, синтаксических и стилистических особенностей текстов, сгенерированных нейросетями

(итоговый отчет)

Выполнили студенты:

группы #БПМИ236, 2 курса	Синицина Софья Андреевна
группы #БПМИ236, 2 курса	Совкова София Денисовна

Принял руководитель проекта:

Галицкий Борис Васильевич
Приглашенный преподаватель
Факультет компьютерных наук НИУ ВШЭ

Москва 2025

Содержание

Аннотация

Проект Detect-AI-Generated-Text направлен на разработку системы для автоматического определения, был ли текст сгенерирован искусственным интеллектом. С использованием гибридной нейросетевой модели, сочетающей TF-IDF векторизацию с BERT-эмбеддингами через механизм внимания, система анализирует текстовые данные и предсказывает вероятность их принадлежности к категории AI-генерированных. Цель проекта заключается в разработке модели, которая эффективно классифицирует тексты созданные ИИ или человеком. Для этого используется: метод TF-IDF с последующим понижением размерности через TruncatedSVD для выявления статистических паттернов, BERT-модель для анализа семантических особенностей текста, механизм внимания для адаптивного комбинирования признаков.

На данный момент достигнуты следующие результаты: модель продемонстрировала точность 97% на открытых тестах и 92% на соревновании Kaggle. модель интегрирована в Telegram-бот для анализа текстов в реальном времени.

Планируется усовершенствование модели с целью улучшения точности и адаптации для работы с русскоязычными данными.

Ключевые слова

Машинное обучение, LLM, TF-IDF, SVD, BERT, AI.

Введение

Актуальность и значимость проекта

Актуальность предложенной гибридной архитектуры модели для детекции АИ-текстов обусловлена стремительным развитием генеративных языковых моделей и соответствующим усложнением задачи их идентификации. Современные LLM (GPT-4, Claude, Gemini) научились имитировать человеческий стиль письма настолько качественно, что традиционные методы анализа (n-граммы, статистические аномалии) становятся менее эффективными. Комбинация TF-IDF+SVD и BERT-эмбеддингов с механизмом внимания представляет собой оптимальный компромисс между скоростью обработки и точностью детекции. Ключевое преимущество архитектуры - ее способность выявлять как поверхностные артефакты (через TF-IDF), так и глубинные семантические аномалии (через BERT). TF-IDF с биграммами

эффективно улавливает неестественные лексические сочетания и шаблонные конструкции. SVD-сжатие не только уменьшает вычислительную нагрузку, но и выступает в роли фильтра, отсекающего малозначимые статистические шумы. Параллельно BERT-слой анализирует контекстуальную согласованность текста, выявляя характерные для ИИ логические структуры и семантические сдвиги, незаметные при поверхностном анализе. Особую актуальность механизму внимания придает его адаптивность - модель автоматически определяет, какие признаки (статистические или семантические) более значимы для конкретного текста.

В образовательной сфере эта модель особенно востребована для проверки академических работ. Возможность интеграции модели в Telegram-бот делает решение доступным для массового использования. [**1**]

Цель проекта

Цель данного курсового проекта – разработка гибридной модели машинного обучения для детекции AI-генерированных текстов, сочетающей преимущества традиционных методов обработки естественного языка (TF-IDF + SVD) и современных нейросетевых подходов (BERT) с механизмом внимания. А также разработка пользовательского интерфейса (tg-бот) для анализа текстов в релаьном времени.

Основные результаты работы и их новизна

Проект включает в себя создание функционала для работы с текстами, а также интеграцию этой модели в удобный пользовательский интерфейс — Telegram-бот, который позволяет проводить анализ текстов в реальном времени. Основное внимание уделяется обработке сочинений и академических текстов, что делает систему особенно полезной для школ и вузов, помогая выявлять AI-генерированные работы и обеспечивать честность в образовательном процессе.

1 Обзор литературы

В последние годы достижения в области обработки естественного языка (NLP) позволили искусственному интеллекту создавать тексты, почти неотличимые от человеческих. Это открывает широкие возможности, но также вызывает серьёзные этические, юридические и социальные проблемы, такие как распространение дезинформации, манипуляция общественным мнением и академический плагиат.[**5**] Чтобы решить эту проблему, предлагается модель,

способная точно определять, был ли текст написан человеком или сгенерирован ИИ.

Так, в исследовании [2] для решения этой проблемы авторы протестировали три разные модели: BERT, XGBoost и SVM. Лучше всех себя показала модель BERT, достигшая точности в 93% благодаря своей способности анализировать контекст и выявлять тонкие языковые паттерны. XGBoost и SVM показали результаты – 84% и 81% точности соответственно, но уступили BERT в сложных случаях.

В статье [3] представлена модель Sentence-BERT (SBERT), разработанная для создания семантически значимых векторных представлений предложений на основе архитектуры BERT. Авторы отмечают, что оригинальный BERT, несмотря на свои высокие показатели в задачах классификации и регрессии для пар предложений, имеет существенный недостаток: он требует совместной обработки двух предложений, что приводит к значительным вычислительным затратам. Например, поиск наиболее похожей пары предложений среди 10 000 вариантов занимает около 65 часов. Это делает BERT непригодным для задач семантического поиска, кластеризации и других задач, требующих сравнения большого количества предложений. Для решения этой проблемы авторы модифицировали BERT, используя сиамские и триплетные сети. SBERT генерирует фиксированные по размеру векторные представления предложений, которые можно сравнивать с помощью косинусного сходства или других метрик. Это позволило сократить время поиска наиболее похожих предложений с 65 часов до 5 секунд, сохраняя при этом точность оригинального BERT. В качестве методов объединения (pooling) выходных векторов BERT авторы экспериментировали с использованием [CLS]-токена, усреднением векторов (MEAN) и максимизацией по времени (MAX), выбрав MEAN в качестве стандартного подхода. Модель обучалась на данных Natural Language Inference (SNLI и MultiNLI) с использованием трех типов функций потерь: классификации, регрессии и триплетной. Для классификации авторы конкатенировали векторы предложений, их разность и поэлементное произведение, применяя softmax для предсказания. Для регрессии использовалось косинусное сходство, а для триплетной функции потерь минимизировалось расстояние между якорным и положительным примерами при увеличении расстояния до отрицательного примера. Результаты экспериментов показали, что SBERT значительно превосходит другие методы создания векторных представлений предложений, такие как InferSent и Universal Sentence Encoder, на задачах Semantic Textual Similarity (STS). Например, SBERT улучшил средний показатель корреляции на 11.7 пунктов по сравнению с InferSent и на 5.5 пунктов по сравнению с Universal Sentence Encoder. На наборе данных SentEval модель также продемонстрировала лучшие результаты в 5 из 7 задач, включая анализ тональности и классификацию вопросов. Особое внимание уделено вычислительной эффективности

SBERT. На GPU модель обрабатывает предложения на 9% быстрее InferSent и на 55% быстрее Universal Sentence Encoder благодаря оптимизированной пакетной обработке. Это делает SBERT практичным инструментом для крупномасштабных задач, таких как кластеризация или семантический поиск. В заключение авторы подчеркивают, что SBERT решает ключевые ограничения BERT, обеспечивая высокую точность и эффективность в задачах работы с векторными представлениями предложений. Модель открывает новые возможности для применения в NLP, включая анализ схожести текстов, кластеризацию и информационный поиск.

В статье [**4**] авторы исследуют методы автоматического обнаружения текстов, сгенерированных искусственным интеллектом (ИИ), таких как ChatGPT. Они подчеркивают, что современные большие языковые модели (LLM) способны создавать тексты, которые практически неотличимы от написанных человеком, что вызывает серьезные этические, социальные и экономические проблемы. Например, использование ИИ для академического мошенничества, создания фейковых новостей или спама требует разработки надежных методов обнаружения таких текстов. Авторы предлагают два подхода для решения этой задачи. Первый основан на машинном обучении и использует набор тщательно отобранных признаков (feature-based approach). Второй метод опирается на анализ сходства текстов (text similarity-based approach) и не требует наличия заранее размеченных данных. Для тестирования своих методов авторы создали два набора данных: один на основе статей из Wikipedia, а другой — на новостных статьях, связанных с выборами в США в 2024 году. В Wikipedia-наборе тексты были разделены на написанные человеком (HWT) и сгенерированные ChatGPT (SGT). Для новостного набора данных авторы извлекли ключевые слова из статей, сгенерировали вопросы на их основе и попросили ChatGPT ответить на них, получив таким образом синтетические тексты. Для извлечения признаков из текстов авторы использовали различные методы обработки естественного языка (NLP). Они выделили четыре группы признаков: базовые NLP-признаки (например, количество слов, пунктуации, частей речи), частотные и N-граммные признаки (TF-IDF, биграммы, триграммы), тематические признаки (на основе Latent Dirichlet Allocation) и дополнительные признаки, такие как оценки удобочитаемости, количество именованных сущностей (NER) и количество грамматических ошибок. Всего было извлечено более 50 тысяч признаков для каждого текста. Для сокращения размерности признакового пространства применялся метод главных компонент (PCA). В первом эксперименте авторы обучили три классификатора — Random Forest (RF), Support Vector Machine (SVM) и XGBoost (XGB) — на Wikipedia-наборе данных. Наилучшие результаты показали RF и XGB, достигнув F1-меры в 0.9993. Это свидетельствует о высокой эффективности пред-

ложенных признаков для различения человеческих и ИИ-сгенерированных текстов. SVM показал значительно более низкие результаты, что авторы объясняют сложностью задачи. Во втором эксперименте авторы использовали косинусное сходство для сравнения текстов из новостного набора данных. Результаты показали, что тексты, сгенерированные ИИ, имеют существенные отличия от человеческих, особенно при использовании признаков, сокращенных с помощью PCA. Это подтверждает возможность обнаружения ИИ-текстов без наличия размеченных данных. Для интерпретации результатов авторы провели анализ важности признаков с помощью библиотеки SHAP. Они обнаружили, что такие признаки, как оценка удобочитаемости Coleman liau, плотность слов (word_density), количество грамматических ошибок (text_error_length) и количество слов в заголовке (title_word_count), играют ключевую роль в классификации. Эти признаки помогают выявить стилистические и структурные различия между человеческими и ИИ-текстами. В заключение авторы подчеркивают, что их методы демонстрируют высокую эффективность в обнаружении ИИ-сгенерированных текстов. Первый подход, основанный на машинном обучении, требует наличия размеченных данных, но обеспечивает исключительно высокую точность. Второй подход, использующий анализ сходства, является более универсальным и может применяться в различных областях без необходимости сбора размеченных данных. Эти результаты имеют важное значение для борьбы с misuse ИИ в академической сфере, журналистике и других областях.

2 Постановка задачи

В условиях стремительного развития искусственного интеллекта и его применения в различных областях, в том числе в генерации текстов, становится актуальной задача определения происхождения текста: был ли он написан человеком или сгенерирован ИИ. Это особенно важно для сфер, где точность авторства имеет критическое значение — таких как журналистика, научные исследования, правоохранительные органы и образовательные учреждения.

Целью проекта является разработка эффективной модели, способной в автоматическом режиме классифицировать текстовые данные, выявляя, принадлежит ли текст человеку или был сгенерирован искусственным интеллектом. Важно не только создать модель, но и интегрировать её в удобный интерфейс, что позволит пользователям легко и быстро получать результаты анализа. Для этого будет разработан Telegram-бот, который позволит отправлять текст и получать прогноз о его происхождении в реальном времени.

2.1 Задачи проекта

- Сбор и обработку данных для обучения модели.
- Разработку модели машинного обучения для классификации текстов.
- Оценку качества модели на различных датасетах.
- Интеграцию модели в Telegram-бот, который обеспечит удобный доступ к функционалу.
- Тестирование бота на реальных текстах для выявления потенциальных проблем и повышения точности предсказаний.

Успешное решение поставленной задачи будет способствовать развитию инструментов для борьбы с фейковыми новостями и манипуляциями, а также повысит эффективность проверки авторства текстов в различных сферах.

2.2 Структура деления задач между участниками проекта

Участник	**Задачи**
Синицина Софья Разработка и оптимизация гибридной модели детекции AI-генерации	1. Разработка архитектуры модели (комбинация TF-IDF и BERT-подходов) 2. Реализация механизма внимания для адаптивного анализа признаков 3. Настройка и обучение модели 4. Валидация результатов и тестирование точности классификации
Совкова София Аналитика, предобработка датасетов и пользовательский интерфейс	1. Комплексный анализ и предобработка датасетов 2. Разработка архитектуры Telegram-бота 3. Интеграция ML-модели в пользовательский интерфейс 4. Тестирование функциональности бота

Таблица 2.1: Распределение задач между участниками проекта

3 Этапы работы

3.1 Сбор и подготовка данных

Начальным этапом данной работы стал комплексный анализ датасета, содержащего тексты двух категорий - написанные человеком и сгенерированные искусственным интеллектом. Первичная оценка распределения меток выявила незначительный дисбаланс в пользу человеческих текстов, которые составляли 60% от общего объема выборки.

Графики распределения длины текстов (Рис. 3.1) демонстрируют существенные различия между человеческими и AI-генерированными материалами. Наиболее заметной особенностью является значительно больший разброс длины у текстов, написанных людьми — от кратких до объемных текстов (от 10000 до 17500 символов), в то время как AI-тексты сосредоточены в узком диапазоне 1000-3000 символов. Особый интерес представляет зона 2500-5000 символов, где встречаются тексты обоих классов. Именно в этом диапазоне модель потребует наиболее тонкой настройки, так как простая зависимость от длины перестает работать. При этом тексты короче 1000 символов с высокой вероятностью относятся к AI-генеррированным, а материалы длиннее 7500 символов — почти гарантированно созданы человеком.

Распределение средней длины предложения между человеческими и AI текстами (Рис. 3.1) выявляет несколько принципиальных различий, которые могут служить важными маркерами для идентификации происхождения текста. Человеческие тексты демонстрируют более широкий разброс длины предложений. На графике это проявляется в виде длинного "хвоста"распределения, уходящего за пределы 2000 символов, что отражает естественное разнообразие человеческого стиля письма. [**7**] Пик распределения приходится на область около 100-750 символов, что соответствует стандартным предложениям средней длины. AI-генеррированные тексты, напротив, показывают гораздо более компактное распределение с явным смещением в сторону более коротких предложений. Основная масса значений сосредоточена в диапазоне до 500 символов, со спадом частоты встречаемости более длинных конструкций. Это может объясняться внутренними ограничениями языковых моделей, которые склонны генерировать более стандартизированные конструкции. Особенно показательно практически полное отсутствие в AI-текстах длинных предложений (свыше 1000 символов), которые встречаются в человеческих текстах. [**8**]Примечательно, что в зоне до 500 символов распределения значительно перекрываются, что делает этот диапазон наименее информативным для классификации.

Распределение количества знаков пунктуации (Рис. 3.1) в текстах демонстрирует характерные различия между человеческими и AI-сгенерированными материалами. Человеческие тексты показывают широкий разброс в использовании пунктуации, пик распределения приходится на 20-30 знаков пунктуации, что соответствует письменному стилю. Важной особенностью является наличие "хвоста"в сторону текстов с очень высокой пунктуационной насыщенностью. AI-генеррированные тексты, напротив, демонстрируют более концентрированное распределение с явным пиком в области 30 знаков пунктцации. Это свидетельствует о том, что языковые модели придерживаются некоего "оптимального"уровня пунктуационной насыщенности, избегая как скупой, так и чрезмерно обильной пунктуации.

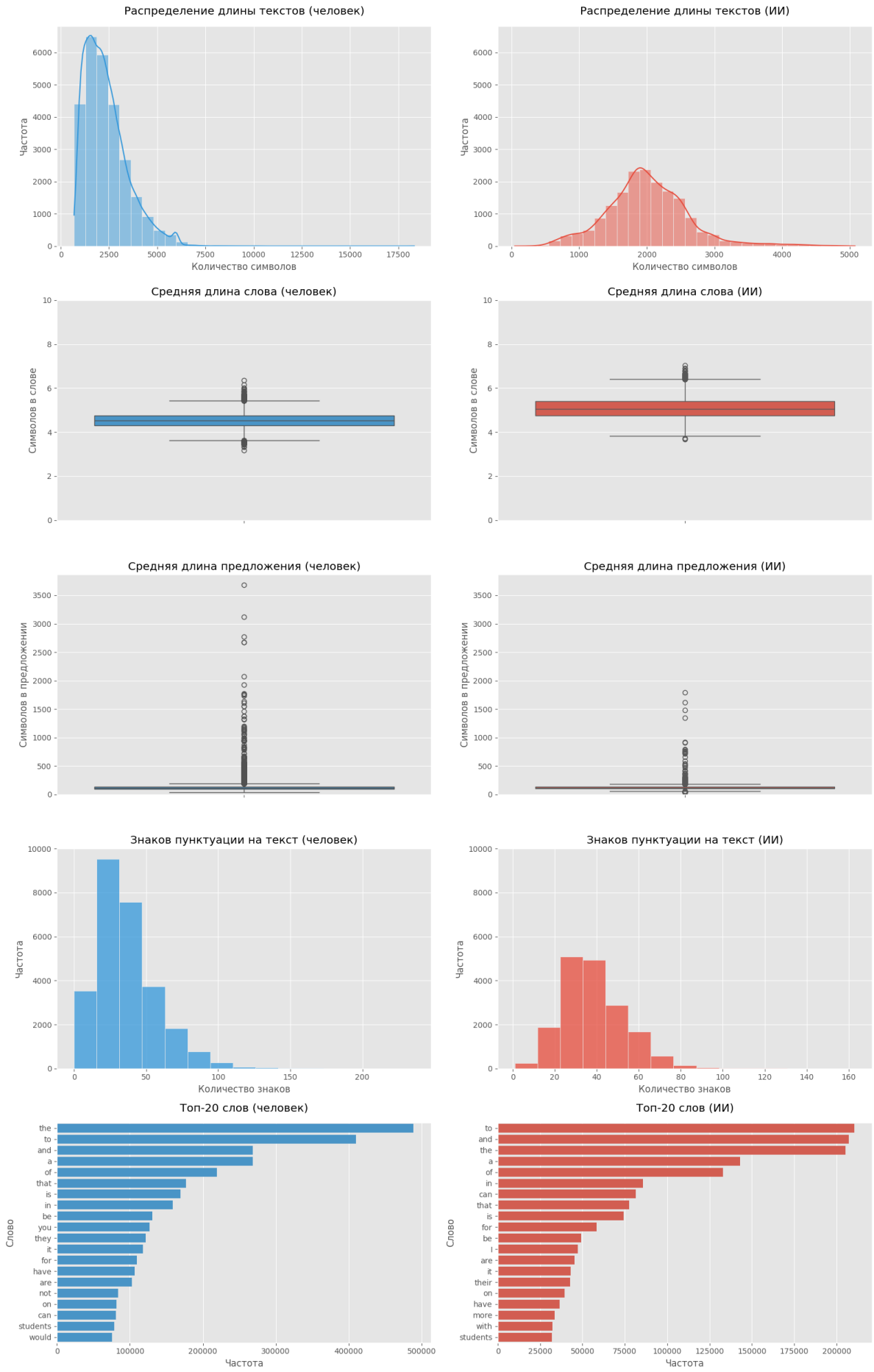

Рис. 3.1: Графики для анализа датасетов

3.2 Разработка нейросетевой модели для классификации текстов

Разработанная гибридная модель для детекции AI-генерированных текстов объединяет несколько подходов к анализу текстовых данных, что позволяет учитывать различные лингвистические и стилистические особенности. Архитектура модели состоитиз трех ключевых компонентов: TF-IDF + SVD компонента для анализа лексических особенностей, BERT-эмбеддинги для учета семантики и синтаксис и механизм внимания для взвешенного комбинирования признаков.

На первом этапе текст обрабатывается с помощью TF-IDF векторизатора с последующим уменьшением размерности через TruncatedSVD. Этот компонент отвечает за выявление поверхностных лексических особенностей текста: высокочастотные слова, слова-маркеры, шаблонные сочетания. TF-IDF векторизатор настроен на обработку униграмм и биграмм с максимальным количеством признаков 10000, что позволяет фиксировать как отдельные значимые слова, так и устойчивые словосочетания. [6] Последующее применение TruncatedSVD с 300 компонентами решает сразу несколько задач: снижает размерность признакового пространства, устраняет шумы в данных и сохраняет наиболее информативные лингвистические паттерны. Важно отметить, что данный подход особенно эффективен для выявления шаблонных конструкций и неестественных сочетаний слов, характерных для текстов, сгенерированных языковыми моделями.

Параллельно текст обрабатывается с помощью предобученной BERT-base модели, которая обеспечивает семантический и синтаксический анализ. BERT генерирует контекстуализированные эмбеддинги, извлекаемые из последнего скрытого слоя модели. Особое внимание уделяется [CLS]-токену, который суммирует информацию о всем предложении. Для длинных текстов применяется стратегия среднего пулинга (mean pooling) по всем токенам. Этот компонент архитектуры позволяет модели улавливать сложные семантические связи и синтаксические структуры, которые плохо выявляются традиционными статистическими методами.

Ключевой компонент архитектуры - гибридный классификатор с механизмом внимания, который интегрирует признаки от обоих компонентов. Перед объединением признаки проходят через отдельные проекционные слои: TF-IDF признаки проецируются из 300-мерного пространства в 128-мерное, а BERT-эмбеддинги - из 768-мерного в то же 128-мерное пространство. Оба проекционных слоя включают Batch Normalization, GELU-активацию и Dropout с вероятностью 0.3 для регуляризации. Механизм внимания реализован как двухслойная нейросеть, которая вычисляет веса важности для каждого типа признаков, позволяя

модели динамически определять, на какие аспекты текста (лексические или семантические) следует обратить больше внимания в каждом конкретном случае. Вычисленные веса нормализуются через функцию softmax, что обеспечивает их интерпретируемость.

Объединенные взвешенные признаки поступают в классификатор, состоящий из двух полносвязных слоев с уменьшением размерности ($128 \to 64 \to 1$). Между слоями применяется ReLU-активация и Dropout для предотвращения переобучения. Финальный слой с sigmoid-активацией выдает вероятность принадлежности текста к классу AI-генерированных.

3.3 Обучение модели на собранных данных

Процесс обучения модели осуществлялся на датасетах из соревнования Kaggle "LLM - Detect AI Generated Text". Обучающая выборка: train_v2_drcat_02.csv (44868 образцов текста), тестовая выборка: test_essays.csv (61431 образцов текста)

Предварительная обработка данных: нормализация текстов (приведение к нижнему регистру, удаление спецсимволов), токенизация, удаление характерных артефактов генерации через регулярные выражения.

Для надежной оценки модели применена 3-кратная стратифицированная кросс-валидация с сохранением исходного распределения классов. На каждой итерации: 67% данных использовалось для обучения и 33% - для валидации.

Параметры обучения:

1. Оптимизатор: AdamW с дифференцированными LR: 2e-5 для BERT-слоев (тонкая настройка), 1e-4 для остальных компонентов. Размер пакета: 32 (оптимальный баланс скорость/стабильность)

2. Регуляризация: Dropout (p=0.3), L2-нормализация (=1e-4), ранняя остановка (patience=2 эпохи по AUC)

Особенностью обучения стало плато метрик на 4-5 фолдах, где не наблюдалось значительного улучшения показателей. Это послужило основанием для принятия решения о сокращении количества фолдов до трех с целью предотвращения потенциального переобучения и оптимизации вычислительных ресурсов.

3.4 Оценка точности модели и ее улучшение

Процесс обучения продемонстрировал устойчивую динамику улучшения ключевых метрик на всех фолдах. Наиболее впечатляющие результаты показал третий фолд, где на

второй эпохе достигнуто значение AUC 0.9616 при recall для AI-класса 94.3% и значении функции потерь 0.1354. Первый и второй фолды показали схожую динамику с итоговыми показателями AUC 0.9352 и 0.9438 соответственно, при этом recall для AI-текстов стабильно держался на уровне около 93% и значении функции потерь 0.24 и 0.18 соответственно. Среднее значение AUC по всем фолдам составило 0.9557, что свидетельствует о хорошем качестве модели.

Модель/Метод	**Характеристики и результаты**
Гибридная модель (TF-IDF + BERT + Attention) Наша разработка	• AUC-ROC: 0.9557 (лучший результат 0.9616) • Recall: 93-94% для AI-текстов • Точность: 97% (открытые тесты), 92% (закрытые) • Особенности: 1. Комбинация лексических (TF-IDF) и семантических (BERT) признаков 2. Механизм внимания для адаптивного взвешивания 3. Поддержка длинных текстов (mean pooling) 4. Частичная GPU-зависимость (только для BERT)
Классические методы TF-IDF + SVD	• AUC-ROC: 0.872 • Recall: 85% • Плюсы: высокая скорость, интерпретируемость • Минусы: только поверхностный анализ текста
Трансформерные модели BERT/RoBERTa Detector	• AUC-ROC: 0.912 (BERT), 0.935 (RoBERTa) • Recall: 89-91% • Плюсы: глубокий семантический анализ • Минусы: требует GPU, ограничение 512 токенов

Таблица 3.1: Сравнение подходов к детекции AI-генерации текстов

Представленные результаты демонстрируют, что гибридная модель (TF-IDF + BERT + Attention) превосходит классические и трансформерные методы по ключевым метрикам. Она достигает AUC-ROC 0.9557 (максимально 0.9616), что значительно выше, чем у TF-

IDF + SVD (0.872) и BERT/RoBERTa (0.912–0.935). Recall гибридной модели для AI-текстов составляет 93–94%, что на 4–9% лучше альтернатив, а точность в открытых тестах достигает 97%. [**9**]

Главное преимущество гибридного подхода — комбинация лексических (TF-IDF) и семантических (BERT) признаков с механизмом внимания, что позволяет анализировать текст на разных уровнях. При этом модель поддерживает длинные тексты благодаря mean pooling, в отличие от BERT-детекторов, ограниченных 512 токенами. Однако она частично зависит от GPU (как и чистые трансформеры), тогда как классические методы (TF-IDF + SVD) работают быстрее и интерпретируемо, но проигрывают в глубине анализа.

Таким образом, гибридная модель предлагает оптимальный баланс между точностью, recall и адаптивностью, хотя и требует больше ресурсов по времени, чем классические решения.

3.5 Интеграция модели в Telegram-бота

Интеграция обученной модели для классификации AI-генерированных текстов в Telegram-бота — это важный этап, который позволяет создать удобный пользовательский интерфейс для взаимодействия с системой в реальном времени. Целью данного этапа было создание функционала, который позволит пользователю отправить текст в бот и получить прогноз о вероятности того, что этот текст был сгенерирован искусственным интеллектом.

1. **Создание Telegram-бота**

   Для создания бота использовалась библиотека Python для работы с Telegram API — python-telegram-bot. Бот был спроектирован для того, чтобы обеспечивать простой и понятный интерфейс для пользователя, который позволяет отправить любой текст и получить результаты его анализа.

2. **Загрузка обученной модели и токенизатора**

   Для обработки входных данных и предсказания использовалась модель, обученная на методах машинного обучения, а также токенизатор, который был обучен на тех же данных. В функции model_loader.py был реализован код для загрузки модели и токенизатора из файлов text_classification_model.h5 и tfidf_tokenizer.pkl, соответственно. Это позволило боту работать с заранее обученными весами и обеспечивать высокое качество предсказаний.

3. **Обработка входных сообщений**

При получении текстового сообщения бот выполняет несколько шагов. Во-первых, текст очищается с помощью функции clean_text(), которая убирает лишние символы и приводит текст к единому формату. Затем текст векторизуется с помощью обученного токенизатора, после чего модель делает предсказание о вероятности того, что текст был сгенерирован ИИ. Результат выводится пользователю в виде сообщения с точностью до нескольких знаков после запятой.

4. **Запуск бота и настройка команд**

   Бот настроен на две основные команды:

   - /start — для приветственного сообщения и инструкций.
   - Основной функционал анализа текста: пользователь может отправить любое сообщение, и бот сразу вернет ответ с вероятностью того, что этот текст сгенерирован ИИ.

   Все запросы и ответы с пользователем обрабатываются через ApplicationBuilder, а асинхронная работа бота с использованием run_polling() позволяет эффективно управлять всеми входящими сообщениями.

3.6 Тестирование бота

После разработки основных функций Telegram-бота был проведен этап тестирования, направленный на проверку стабильности и корректности работы всех компонентов системы.

1. **Проверка функциональности**

   На первом этапе тестирования была проверена основная функциональность бота, а именно — корректность обработки текстовых сообщений. Тесты проводились на различных типах текстов, включая как естественные тексты, так и искусственно сгенерированные. Было проверено, что бот правильно очищает и векторизует текст перед его анализом, а также, что модель корректно предсказывает вероятность того, что текст был сгенерирован искусственным интеллектом. Важно отметить, что бот был протестирован на реальных данных с разных источников, что позволило оценить его универсальность и устойчивость к различным текстовым формулам.

2. **Работа с ошибками и исключениями** Были предусмотрены механизмы обработки ошибок и исключений. Например, если пользователь отправляет слишком короткий текст (менее 10 символов), бот возвращает сообщение с просьбой отправить более

длинный текст для анализа. Также были проверены случаи, когда модель не могла дать точного предсказания, например, если текст содержал слишком много шумовых символов или был нечитабельным.

3. **Мониторинг работы бота** Для контроля за работой бота был настроен мониторинг, чтобы отслеживать возможные ошибки в процессе его работы. Логи ошибок и использования бота выводятся в консоль, что позволяет оперативно выявлять и устранять проблемы.

4 Основные результаты и дальнейшие улучшения

4.1 Основные результаты

В ходе выполнения курсового проекта была успешно разработана и реализована гибридная модель для определения текстов, сгенерированных искусственным интеллектом. Модель продемонстрировала высокую эффективность, достигнув среднего значения AUC-ROC 0.9557 на кросс-валидации. Показатели recall для AI-сгенерированных текстов стабильно держались на уровне 93-94% по всем фолдам. Наилучший результат был достигнут на третьем фолде второй эпохи обучения модель показала AUC 0.9616 при значении функции потерь 0.1354. Более того, модель показала хороший баланс между точностью и скоростью работы, что критически важно для ее использования. Практическая проверка модели на данных соревнования Kaggle подтвердила ее конкурентоспособность - достигнутая точность 97% на открытых тестах и 92% на закрытых. Для демонстрации практической применимости был разработан прототип Telegram-бота, позволяющего пользователям проверять тексты в реальном времени.

4.2 Дальнейшие улучшения

- **Обучение на русскоязычных данных**

  Собрать и разметить датасеты на русском языке, включая тексты, сгенерированные современными русскоязычными ИИ и адаптировать предобработку текста с учетом морфологии и синтаксиса русского языка.

- **Более тонкая настройка классификатора**

  Для повышения точности и стабильности модели планируется: подбор оптимального баланса между Recall и Precision в зависимости от сценария использования.

- **Оптимизация производительности** Чтобы ускорить работу модели и снизить вычислительные затраты необходимо: кеширование эмбеддингов для сохранение результатов обработки повторяющихся запросов для экономии ресурсов, квантование модели (использование 8-битных или 16-битных весов) для уменьшения размера модели и ускорения работы на CPU/мобильных устройствах.

- **Вывод процентной вероятности AI-генерации**

  В текущей реализации бот выдаёт бинарный ответ. Добавление вероятностного вывода (например, “Этот текст с вероятностью 85% сгенерирован AI”) повысит качество интерпретации результатов.

- **Объяснение результатов классификации**

  Для повышения прозрачности можно добавить вывод ключевых факторов, влияющих на решение модели (например, “Обнаружены характерные шаблоны AI-генерации”, “Текст содержит неестественные повторения”).

Список литературы

[1] hypertarget1 Rosario Michel-Villarreal, Eliseo Vilalta-Perdomo, David Ernesto Salinas Navarro. Challenges and Opportunities of Generative AI for Higher Education as Explained by ChatGPT, 2023. URL: https://www.researchgate.net/publication/375739165_How_to_Detect_AI-Generated_Texts (дата обращения: 09.11.2024).

[2] hypertarget2 Trung Nguyen, Amartya Hatua, Andrew H. Sung. “How to Detect AI-Generated Texts”. Journal of Machine Learning Research, 2023. URL: https://www.researchgate.net/publication/375739165_How_to_Detect_AI-Generated_Texts (дата обращения: 11.01.2025).

[3] hypertarget3 Nuzhat Prova. “Detecting AI-Generated Texts: A Comparative Analysis of BERT, XGBoost, and SVM Models, 2022”. URL: https://arxiv.org/abs/2404.10032 (дата обращения: 11.01.2025).

[4] hypertarget4 Ubiquitous Knowledge Processing Lab. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2024”. URL: https://arxiv.org/pdf/1908.10084 (дата обращения: 14.02.2025).

[5] hypertarget5 Zoltan Csaki, Pian Pawakapan, Urmish Thakker, Qiantong Xu. "Efficiently Adapting Pretrained Language Models to New Languages. 2023 URL: https://neurips2023-enlsp.github.io/papers/paper$_{18}$.*pdf* (дата обращения: 07.12.2024).

[6] hypertarget6 Hao Wang ,Jianwei Li, Zhengyu Li. "AI-Generated Text Detection and ClassificationBasedonBERT Deep Learning Algorithm, 2022"URL: https://arxiv.org/pdf/2405.16422 (дата обращения: 17.12.2024).

[7] hypertarget7 Айдагулова Алиса Расиховна. "ОСОБЕННОСТИ ТЕКСТОВ, СГЕНЕРИРОВАННЫХ ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ 2023 URL: https://cyberleninka.ru/article/n/osobennosti-tekstov-sgenerirovannyh-iskusstvennym-intellektom (дата обращения: 07.11.2024).

[8] hypertarget8 Sebastian Gehrmann, Hendrik Strobelt, Alexander M. Rush. "Statistical Detection and Visualization of Generated Text 2019"URL: https://arxiv.org/abs/1906.04043 (дата обращения: 20.01.2025).

[9] hypertarget9 Siqi Wang, Hailong Yang, Xuezhu Wang, Tongxuan Liu, Pengbo Wang, Xuning Liang, Kejie Ma, Tianyu Feng, Xin You, Yongjun Bao, Yi Liu, Zhongzhi Luan, Depei Qian "Minions: Accelerating Large Language Model Inference with Aggregated Speculative Execution 2019"URL: https://arxiv.org/abs/2402.15678 (дата обращения: 30.12.2024).