Факультет компьютерных наук

Образовательная программа “Прикладная математика и информатика”

Москва 2025

Анализ и классификация текстов с целью выявления авторства искусственного интеллекта

Исследование лексических, синтаксических и стилистических особенностей текстов, сгенерированных нейросетями

Выполнили:
студентка БПМИ236 **Синицина Софья Андреевна**
студентка БПМИ236 **Совкова София Денисовна**

Руководитель: **Галицкий Борис Васильевич**

Актуальность работы

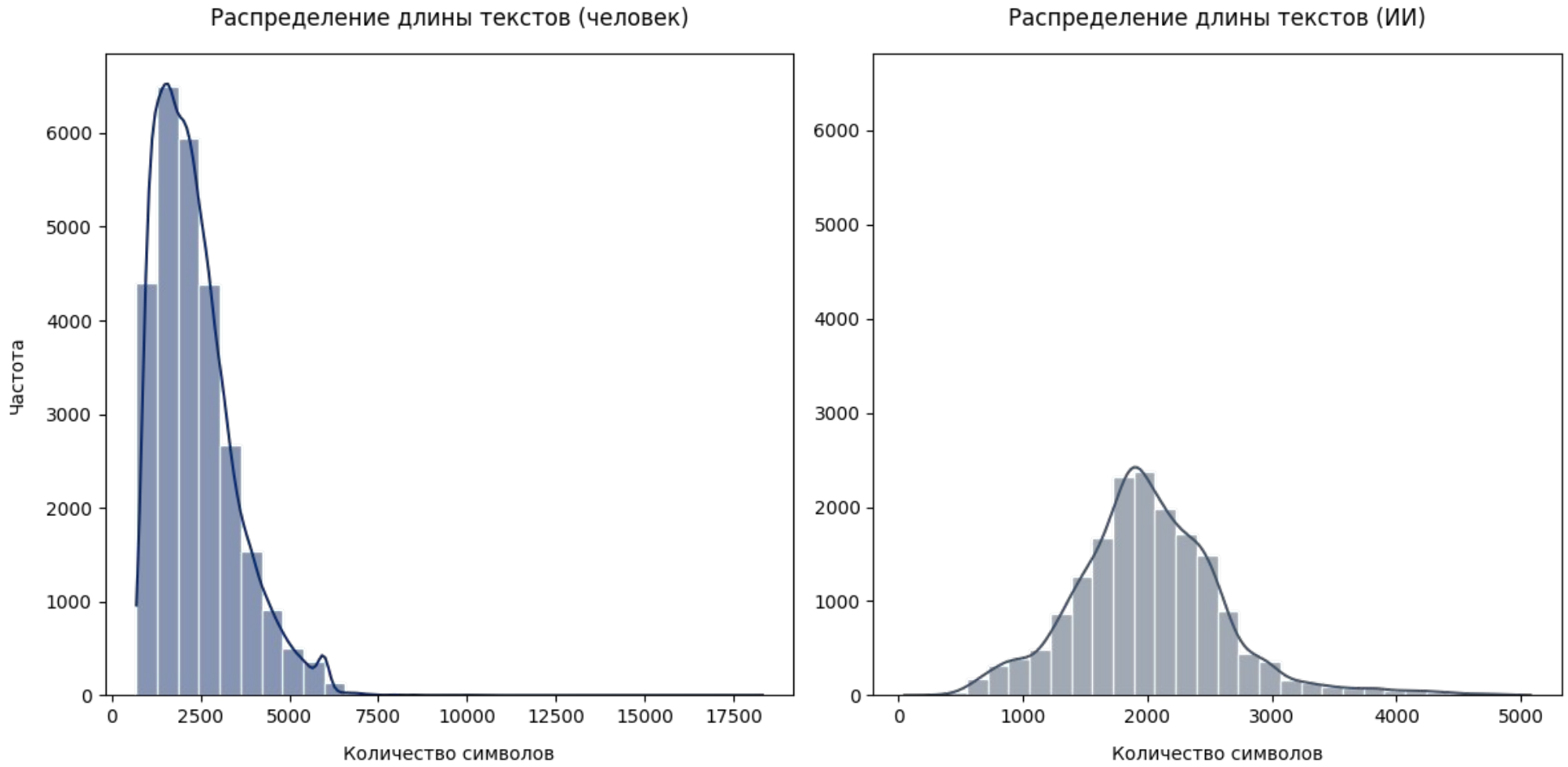
Проблема	Решение
Современные LLM создают тексты, почти неотличимые от человеческих	Создание высокоточной гибридной модели, позволяющей определять авторство
Традиционные методы анализа (n-граммы, статистика) теряют эффективность	Сочетание лексического анализа (шаблоны, статистические аномалии) и семантическую глубину (контекст, логические структуры)
Ограничение на количество запросов в сервисах антиплагиата и необходимость в поиске бесплатного и без регистрации	Интеграция в модели в Telegram для анализа текстов в реальном времени

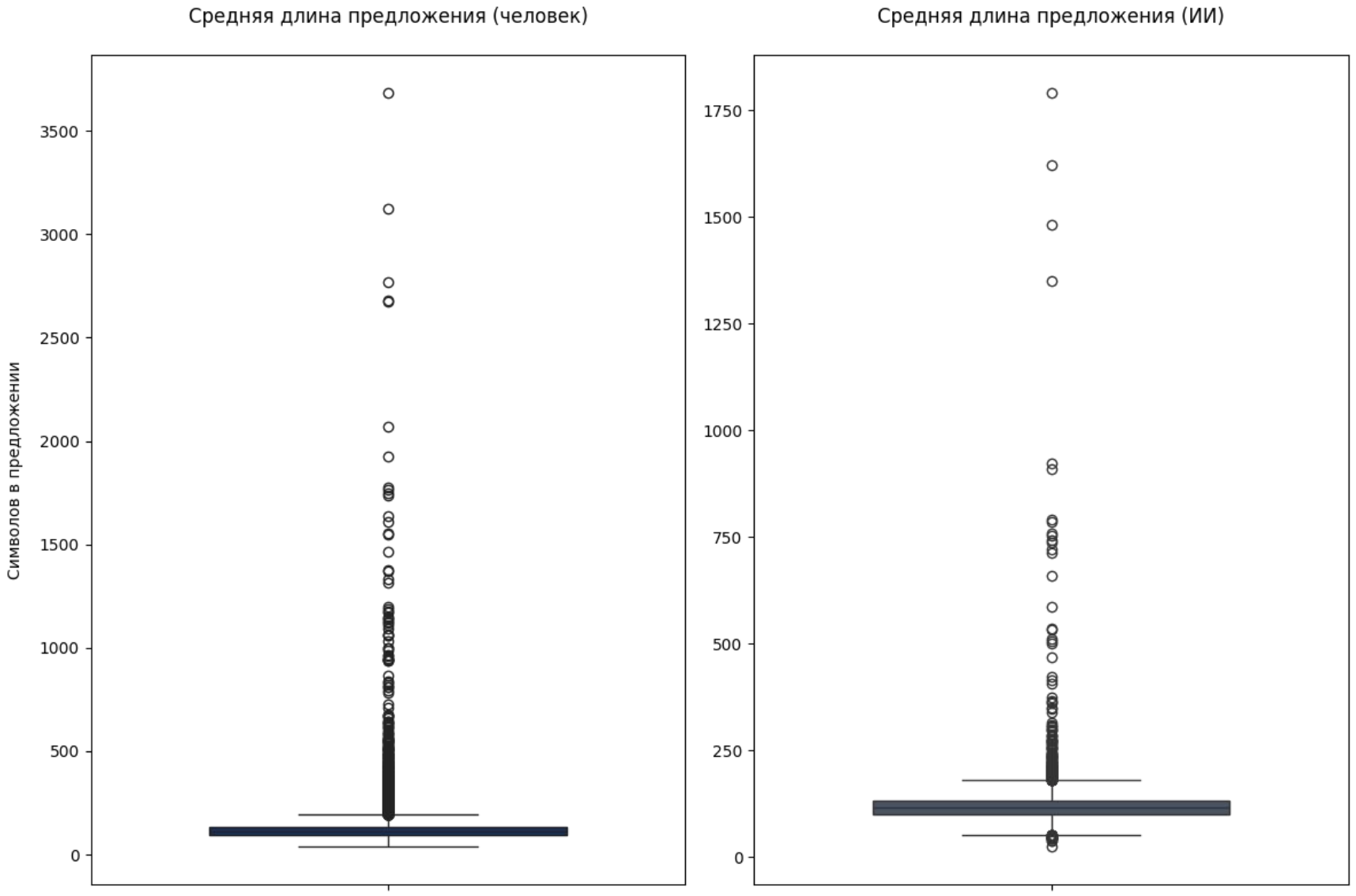
Цель работы – разработать гибридную модель машинного обучения для детекции AI-сгенерированных текстов.

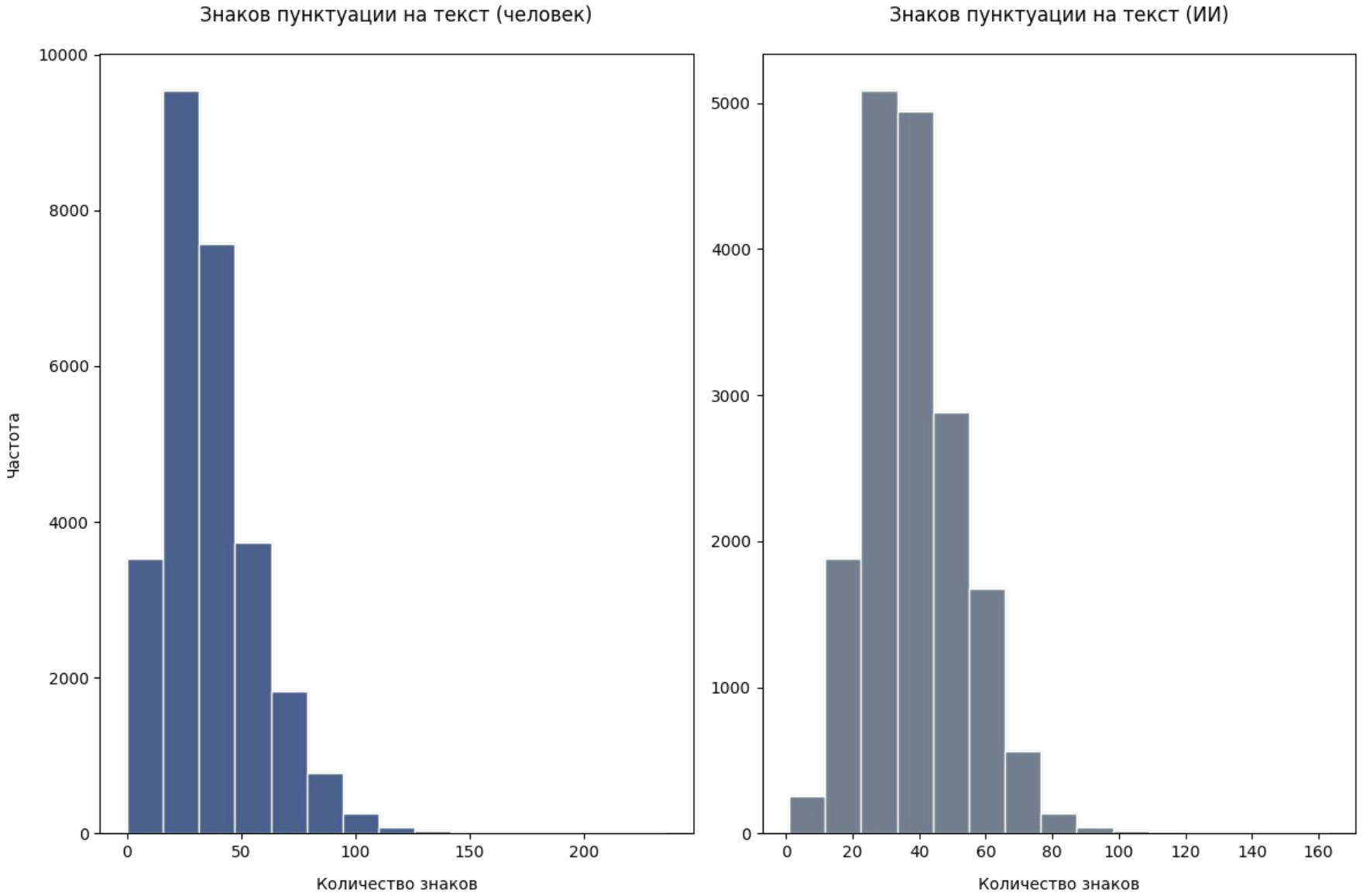
Задачи

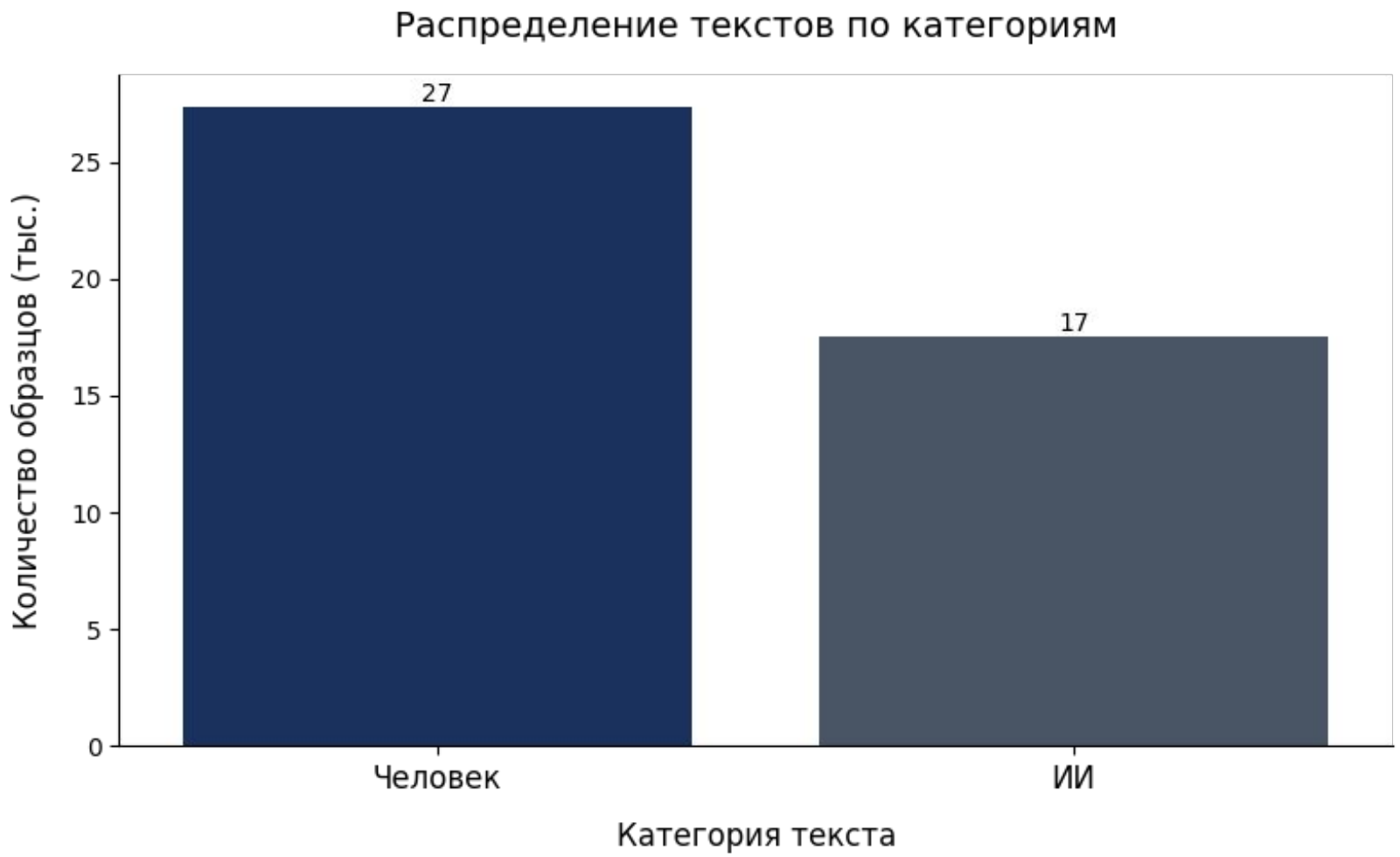
1) Собрать и обработать данные для обучения модели.
2) Разработать модель машинного обучения для классификации текстов.
3) Оценить качество модели на различных датасетах.
4) Интегрировать модель в Telegram-бот, который обеспечит удобный доступ к функционалу.
5) Провести тестирование бота на реальных текстах для выявления потенциальных проблем и повышения точности предсказаний.

Анализ датасетов

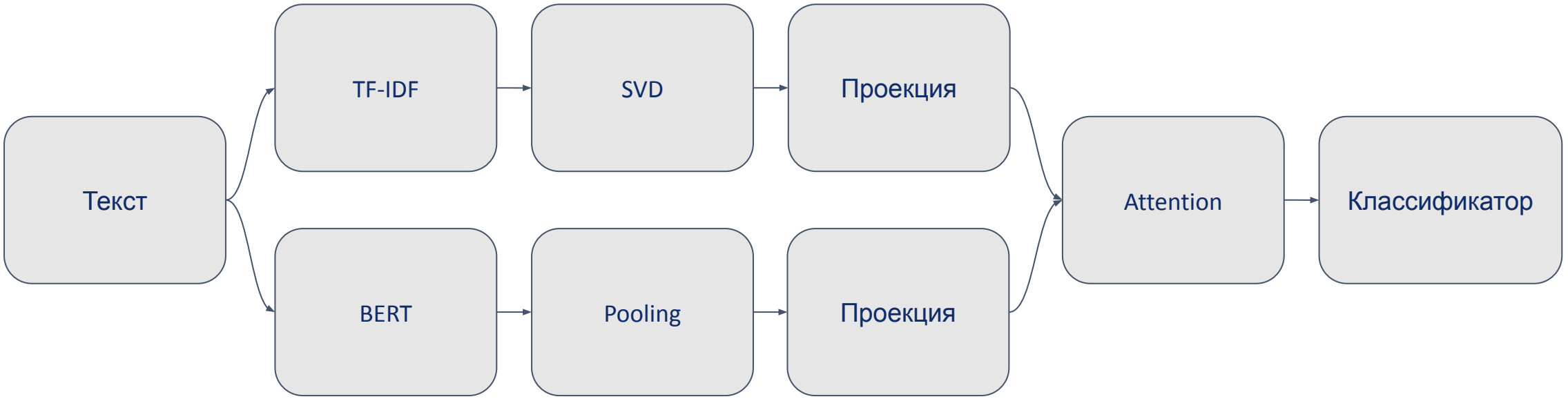



Средняя длина предложения (человек)
Символов в предложении
0
500
1000
1500
2000
2500
3000
3500
Средняя длина предложения (ИИ)
0
250
500
750
1000
1250
1500
1750


Знаков пунктуации на текст (человек)
Частота
Количество знаков
0
2000
4000
6000
8000
10000
0
50
100
150
200
Знаков пунктуации на текст (ИИ)
Количество знаков
0
1000
2000
3000
4000
5000
0
20
40
60
80
100
120
140
160


Распределение текстов по категориям
Количество образцов (тыс.)
0
5
10
15
20
25
27
17
Человек
ИИ
Категория текста

Архитектура модели



Ветвь TF-IDF

$$\text{TF-IDF}(t,d)=\text{TF}(t,d)\times\text{IDF}(t)$$

$$\text{TF}(t,d) = \frac{\text{Количество вхождений } t \text{ в } d}{\text{Общее число слов в } d}$$

$$\text{IDF}(t) = \frac{\text{Количество вхождений } t \text{ в } d}{\text{Количество документов, содержащих термин } t}$$

$$B_{Nx300} = AV_k\ , \quad A_{Nx1000} = U_k\Sigma_k V_k^T\ , \quad k = 300$$

U_k — левые сингулярные векторы ($N \times 300$),
Σ_k — диагональная матрица сингулярных чисел (300×300),
V_k^T — правые сингулярные векторы (300×10000)

Ветвь Bert


N текстов
BERT
Токенизация
– ID токенов
– Маска токенов
BERT
Кодирование
Тензор $C_{Nx128x768}$
N - число текстов
128 - длина последовательности
768 - размер эмбеддинга каждого токена
Выбор [CLS]-токена
Тензор D_{Nx768}
N — количество текстов в батче
768 — размерность эмбеддинга BERT

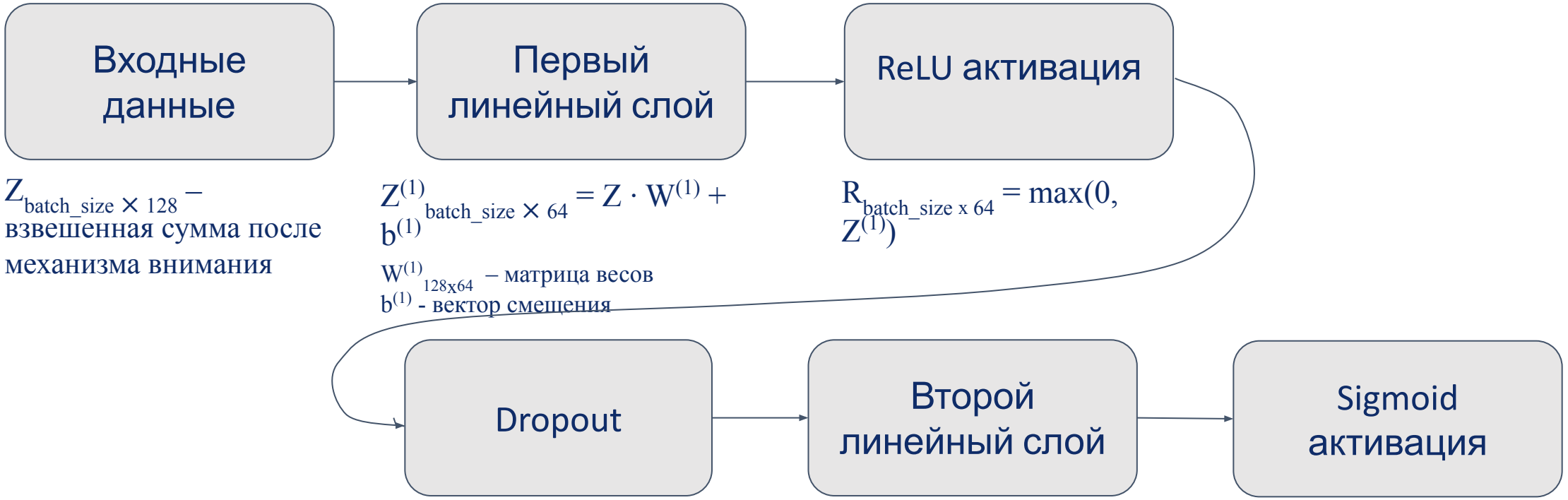
Проекция

Ветвь	TF-IDF	Bert
Изменение размерности	300D -> 128D	768D -> 128D
Формула	$\text{Proj}_{\text{TF-IDF}} = \text{GELU}(W_{\text{TF-IDF}} \cdot B + b_{\text{TF-IDF}})$ B - плотная матрица после SVD (N × 300) $W_{\text{TF-IDF}}$ - веса проекции (300 × 128)	$\text{Proj}_{\text{Bert}} = \text{GELU}(W_{\text{Bert}} \cdot D + b_{\text{Bert}})$ D - тензор [CLS]-токенов (N × 768) $W_{\text{TF-IDF}}$ - веса проекции (768 × 128)

Механизм внимания

Объединение признаков	$H = [Proj_{TF\text{-}IDF} \,||\, Proj_{Bert}] \;\; (batch\_size \times 2 \times 128)$ $batch\_size = 32$
Вычисление весов внимания	$\alpha = softmax(W_\alpha \cdot H + b_\alpha), \quad \alpha \in R^{N \times 2}$ W_α - обучаемые веса (128×1) α - определяет важность каждой ветви (TF-IDF vs Bert)
Взвешенная сумма	$Z = \alpha_1 \cdot Proj_{TF\text{-}IDF} + \alpha_2 \cdot Proj_{BERT} \quad (batch\_size \times 128)$

Классификатор



$Z_{batch\_size \times 128}$ – взвешенная сумма после механизма внимания

$Z^{(1)}_{batch\_size \times 64} = Z \cdot W^{(1)} + b^{(1)}$

$W^{(1)}_{128x64}$ – матрица весов

$b^{(1)}$ - вектор смещения

$R_{batch\_size\ x\ 64} = \max(0, Z^{(1)})$

$A_{drop} = R \odot M$

$M \in \{0, 1\}^{32x64}$

$P(M_{ij} = 1) = 0,7$

$Z^{(2)}_{batch\_size \times 1} = A_{drop} \cdot W^{(2)} + b^{(2)}$

$W^{(2)}_{64x1}$ – матрица весов

$b^{(1)}$ - скалярное смещение.

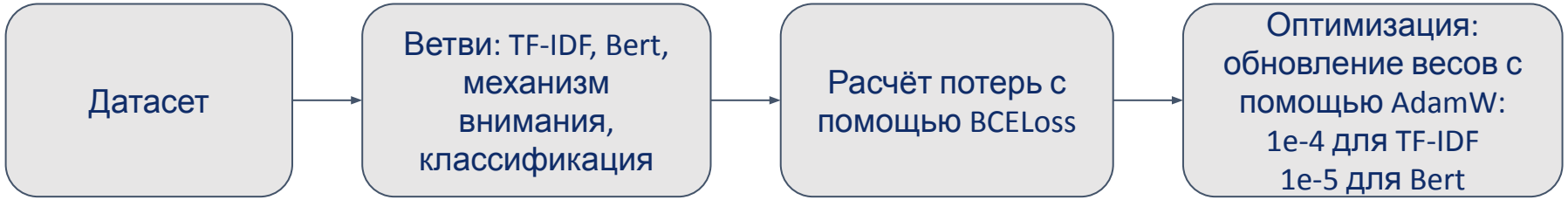
$P = \sigma(Z^{(2)}) = 1 / (1 + e^{-Z(2)})$,

$P \in R^{batch\_size \times 1}$

Обучение модели

Подготовка данных: разделение на тренировочные и валидационный фолды с помощью StratifiedKFold.

Цикл обучения одной эпохи:



Валидация после каждой эпохи:

- Метрики AUC-ROC, F1, Accuracy, Recall (AI)
- Ранняя остановка: Если AUC не растёт 2 эпохи происходит остановка обучения и сохраняются веса модели с лучшим AUC.

Обучение модели

Метрики на кросс-валидации (3 фолда)

Фолд	Лучший AUC	Recall (AI)	Loss
1	0.9352	93.1%	0.240
2	0.9438	93.5%	0.180
3	**0.9616**	**94.3%**	**0.135**
Среднее	0.9557 ± 0.008	93.6%	0.185

Динамика обучения (на примере 3-го фолда)

Эпоха	Train Loss	Val AUC	Val Recall (AI)
1	0.452	0.9412	92.1%
2	**0.210**	**0.9616**	**94.3%**
3	0.185	0.9580	93.8%

Время: $O(T \cdot N_{train} \cdot d^2 + N_{val} \cdot d))$

$d = 128$, T - число эпох

Память: $O(K \cdot (d^2 + N_{batch} \cdot d))$

K - количество фолдов, batch_size = 32

Сравнение с аналогами

Модель/метод	Характеристики	Преимущества	Недостатки
Гибридная модель (TF-IDF + BERT + Attention)	• AUC-ROC: 0.9557 • Recall: 93–94% для AI-текстов	• Комбинация лексических и семантических признаков • Механизм внимания для адаптивного анализа • Высокая точность и recall	• Зависимость от GPU для BERT-компонента • Высокие вычислительные затраты
Классические методы (TF-IDF + SVD)	• AUC-ROC: 0.872 • Recall: 85%	• Высокая скорость работы • Низкие требования к ресурсам • Простота реализации	• Поверхностный анализ текста • Низкая точность для сложных случаев • Не учитывает семантику
Трансформерная модель (BERT)	• AUC-ROC: 0.912 (BERT) • Recall: 89–91%	• Высокое качество анализа контекста • Хорошая адаптивность к разным стилям текста	• Ограничение на длину текста (512 токенов) • Требует GPU для эффективной работы • Высокие вычислительные затраты
Методы на основе N-грамм и статистики	• Точность: 80–85%	• Быстрая обработка • Эффективность для простых текстов	• Низкая точность для современных LLM • Не учитывает семантические особенности • Чувствительность к шумам

Telegram-бот

- Инструмент для преподавателей, редакторов, исследователей
- Анализ текстов в режиме реального времени
- Простой и интуитивный интерфейс
- Доступ с любого устройства через Telegram
- Возможность дальнейшей доработки, добавление новых моделей классификации

Ссылка на бота

Результаты

- Проведен глубокий анализ особенностей текстов, сгенерированных нейросетями
- Создана и реализована гибридная модель для определения текстов, сгенерированных искусственным интеллектом.
- Достигнуто среднее значение AUC-ROC на кросс-валидации: **0.9557**, показатели recall для AI-сгенерированных текстов: **93-94%** по всем фолдам.
- Точность модели составила: **97%** на открытых тестах, **92%** – на закрытых на соревновании Kaggle
- Разработан Telegram-бот для проверки текстов в реальном времени.

Дальнейшие улучшения проекта

- Обучение на русскоязычных данных
- Более тонкая настройка классификатора
- Оптимизация производительности
- Вывод процентной вероятности AI-генерации
- Объяснение результатов классификации