САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Сыдыгалиева Бегаим Нурбековна

Применение методов теории игр и машинного обучения в анализе текстов

Направление: 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа: СВ.5005. «Прикладная математика, фундаментальная информатика и программирование»
Профиль: «Системный анализ, исследование операций и управление»

Научный руководитель:
профессор, кафедра математической теории игр и статистических решений, д.т.н. Буре Владимир Мансурович

Рецензент:
доцент, кафедра математического моделирования энергетических систем, к.ф. - м.н. Свиркин Михаил Владимирович

Санкт-Петербург
2025 г.

Содержание

Введение

Текст как средство передачи знаний и идей играет ключевую роль в различных сферах человеческой деятельности. В условиях стремительного роста объема информации возникает необходимость в эффективных методах анализа и обработки текстовых данных. Одним из важных аспектов в этой области является оценка сложности текстов, что особенно актуально в сфере преподавания русского языка как иностранного. Точная оценка сложности позволяет адаптировать учебные материалы под уровень подготовки целевой аудитории, что способствует повышению эффективности образовательного процесса .

Задача классификации текстов относительно сложности не является новой, исследования данной проблемы представлены в статьях [][14]. Однако большая часть работ изучает аспекты английского языка, а решения, представленные для русского, основаны на классических методах машинного обучения и рассматривают статистические признаки текста. Как следствие, в связи с особенностями русского языка, игнорируются одни из ключевых характеристик: синтаксические и морфологические.

Теория игр предоставляет математический инструментарий для моделирования задач. Машинное обучение, в свою очередь, предоставляет возможности для автоматизации процесса обработки больших объемов текстовых данных. NLP(Natural Language Processing)-технологии позволяют создавать модели, способные оценивать сложность текстов на основе их лексических, синтаксических и семантических характеристик, что помогает выявлять скрытые закономерности. Таким образом, сочетание этих методов открывает новые перспективы для автоматизированного анализа текстов.

В настоящей работе представлены способы решения задачи для учебных текстов с экспертной лингвистической разметкой. Анализируются ограничения традиционных методов оценки сложности и обосновывается выбор альтернативных подходов. Описан процесс построения векторов признаков с использованием игровых моделей и машинного обучения. Представлены эксперименты, включающие сравнение точности моделей CatBoost, SVM, RNN и Transformers(BERT) на датасетах с CEFR-разметкой, а также интерпретация вклада признаков через Шепли-значения и использование данной идеи для основы гибридного подхода.

Результаты исследования демонстрируют, что комбинирование методов открывает перспективы для создания гибких систем анализа текстов, сочетающих математическую строгость с мощностью нейросетей.

Постановка задачи

Пусть заданы:

1. Множество учебных текстов:

$$\mathcal{T} = \{T_i\}_{i=1}^{M}, \quad T_i = \{t_{i,j}\}_{j=1}^{N_i}$$

где:

M — общее количество текстов,

$N_i = |T_i|$ — длина i-го текста,

$T_i = (t_{i,1}, \ldots, t_{i,N_i})$ – i-й текст, представленный как последовательность токенов,

$t_{i,j} \in \mathcal{V}$ — лингвистическая аннотация j-й единицы текста (токена).

2. Множество уровней CEFR(Common European Framework of Reference) - система уровней владения иностранным языком, используемая в Европейском союзе

$$\mathcal{Y} = \{\text{A1}, \text{A2}, \text{B1}, \text{B2}, \text{C1}\}$$

3. Подмножество с лингвистической разметкой

$$\mathcal{D} = \{(T_i, y_i)\}_{i=1}^{L} \subset \mathcal{T} \times \mathcal{Y}$$

Требуется построить отображение $f : \mathcal{T} \to \mathcal{Y}$, минимизирующее функционал ошибки:

$$\mathcal{L}(f) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(T,y) \in \mathcal{D}_{\text{test}}} \mathbb{I}[f(T) \neq y] \to \min_f$$

где:

$\mathcal{D}_{\text{test}} \subset \mathcal{D}$ – тестовая часть выборки

y – истинная метка уровня CEFR для текста T

$\mathbb{I}[\cdot]$ – индикаторная функция:

$$\mathbb{I}[f(T) \neq y] = \begin{cases} 1, & \text{если } f(T) \neq y \\ 0, & \text{если } f(T) = y \end{cases}$$

$\mathcal{D}_{\text{test}} \subset \mathcal{D}$ – тестовая выборка

Цель работы: Разработка и сравнительный анализ методов оценки сложности учебных текстов с использованием подходов теории игр и машинного обучения, а также выявление их эффективности в контексте задач анализа текстовой информации.
Критерии качества модели: Модель должна максимизировать одну из следующих метрик:

1. Accuracy (доля верных предсказаний):

$$\text{Accuracy} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}(y_k = \hat{y}_k)$$

2. F1-мера (гармоническое среднее precision и recall):

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

где:

- Precision (точность) — доля истинно положительных случаев среди всех положительных предсказаний:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall (полнота) — доль истинно положительных случаев, которые были верно идентифицированы:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Обзор литературы

Для изучения способов применения методов теории игр в анализе текстов была использована следующая литература: [1], [2], [3]. В статье [1] представляется способ классификации текстов относительно сложности с использованием игр голосования. Рассматривается способ векторного представления текста с помощью индексов Шепли и Банцафа с последующей кластеризацией. В [2] рассказывается вся необходимая теория. В [3] описаны различные области приложения игр с коалиционной структурой.
Для реализации и изучения методов машинного обучения была использована следующая литература: [10, 11, 12, 13, 14]. Статьи содержат описания применения различных методов классификации текстоы, рассматриваются подходы и проводится сравнительный анализ. Выявляются преимущества нейросетевых подходов перед классическими статистическими. Для выбора необходимых признаков для анализа текста была рассмотрена следующая литература: [4, 5, 6, 7, 8, 9] - исследовательские работы лингвистов

Глава 1. Теоретико-методологические основы исследования

1.1 Критерии оценки сложности текстов

В качестве критериев рассматриваются количественные параметры оценки сложности текста на разных уровнях языка: фонологическом, морфологическом, синтаксическом и лексическом. Кроме того, рассмотрены статистические показатели, такие как индексы удобочитаемости (тест Флеша-Кинкейда, FRE).
Классичесские методы, признаки, разработанные экспертами-лингвистами на основе анализа языковых структур:

1.1.1 Статистические метрики

[4]
Индекс удобочитаемости Флеша (FRE): Метрика, показывающая сложность восприятия текста по 100-балльной шкале (0 – сложный текст для выпускников профильных вузов, 100 – легкий текст для обучающихся начальной школы).

$$FRE = 206.835 - 1.015 \left(\frac{\text{total words}}{\text{total sentences}} \right) - 84.6 \left(\frac{\text{total syllables}}{\text{total words}} \right)$$

Где:

`total words` – общее количество слов в тексте.

`total sentences` – общее количество предложений в тексте.

`total syllables` – общее количество слогов в тексте

Тест Флеша-Кинкейда (FKGL): Метрика, определяющая потенциальный уровень подготовки обучающихся, для которых предназначен текст.

$$FKGL = 0.39 \left(\frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \left(\frac{\text{total syllables}}{\text{total words}} \right) - 15.59$$

Индекс SMOG(Simple Measure of Gobbledygook): Метрика, определяющая сколько лет обучения необходимо для понимания текста.

$$\text{SMOG} = 1.043 \times \sqrt{\text{SMOG\_complex\_words} \times \left(\frac{30}{\text{total\_sentences}} \right)} + 3.1291$$

где:

- SMOG_complex_words – количество слов с 3 и более слогами
- total_sentences – общее количество предложений в тексте

Следующая группа признаков является интуитивно ясной, так как коррелирует с лингвистическими параметрами и является в некотором смысле статистической аппроксимацией лингвистических особенностей:

1.1.2 Фонологический уровень

Количество слогов в слове и средняя длина слова влияют на скорость чтения, тем самым непосредственно воздействуют на восприятие информации.
Оценка сложности на фонологическом уровне включает анализ структуры слов и предложений:

– Количество слогов.

– Среднее количество слогов на слово.

– Средняя длина предложения.

1.1.3 Морфологический уровень

– Доля сложных слов.
Критерии:

- $>$ 3 слогов - Длинные слова ухудшают читаемость.
- длина слова $>$8 символов.

– Количество глаголов в условном наклонении.

– Использование пассивных конструкций.
Данный признак характерен для научного стиля.

– Частотность словоизменительных форм (причастий, деепричастий).

Особую категорию составляют признаки, характеризующие содержательно-структурную сложность текста и описывающие его на более глубоком уровне:

1.1.4 Синтаксический уровень

На синтаксическом уровне анализируются:

– Связи слов в предложении.

– Общее количество слов и предложений.
Длинные предложения содержат больше синтаксических связей.

– Средняя длина слов и предложений.

– Количество сложных конструкций (придаточных предложений).

– Глубина синтаксического дерева.

1.1.5 Лексический уровень

[9] Основными показателями сложности на лексическом уровне являются:

– Частеречное разнообразие.

– Коэффициент лексического разнообразия (Type Token Ratio, TTR).

$$TTR = \frac{\text{Количество уникальных слов (Types)}}{\text{Общее количество слов (Tokens)}}$$

TTR принимает низкое значение при большой повторяемости слов, что влечет за собой принадлежность текста к уровню A1–A2.
Высокий TTR характеризует разнообразие лексики, что свойственно текстам уровеня C1–C2.

1.1.6 Семантический уровень

Темы текстов обуславливаются исходя из возрастных особенностей обучающихся и их уровня владения иностранным языком.[14]
В качестве признака рассматривается тематическая сложность. В реализации алгоритмов представляется через векторные представления текстов (эмбеддингов), полученные с помощью модели BERT(Bidirectional Encoder Representations from Transformers)

– Семантическая плотность:
Метод: Стандартное отклонение BERT-эмбеддингов

– CLS-эмбеддинг
Метод: Векторизация через RuBERT-tiny2 первого токена [CLS]. Высокая дисперсия будет означать разнообразие тем, что обычно характеризует сложные тексты.

1.1.7 Прагматический уровень

– Использование модальных глаголов.
Задают тексту абстрактность, грамматическую сложность.

1.1.8 Стилистические признаки

– Идиомы.
Знание и понимание идиом и метафор невозможно без глубокого погружения в культурный контекст, что тяжело в начальных этапах изучения.

– Термины.

1.1.9 Признаки связности

– Кореферентные цепочки - последовательности упоминаний в тексте, которые относятся и отсылают к одному и тому же объекту.
Данный признак характерен для академических текстов.

– Лексические повторы.

– Доля местоимений.

Для вычислений были использованы библиотеки: Razdel, Transformers, Torch, Natasha(NER - Named Entity Recognition для русского языка), DeepPavlov, пользовательские функции
BERT (Bidirectional Encoder Representations from Transformers) - большая языковая модель, нейронная сеть, основанная на архитектуре трансформера. Является автокодировщиком и предназначена для предобучения векторных представлений языка с целью дальнейшего использования.
Далее в работе используется и для самой классификации, архитектура модели будет описана подробнее

1.2 Предобработка данных

Данные собраны из учебников, рекомендованных РКИ и также из ресурсов открытого корпуса Taiga[15]

Предобработка данных

Приведение символов к одному регистру: для унификации текста.

Токенизация: разбиение текста на токены (слова, знаки препинания и т.д.).

Тегирование частей речи: определение частей речи в каждом предложении для применения грамматических правил.

Лемматизация и стемминг:

Стемминг: грубое приведение слов к корням путем обрезания суффиксов.

Лемматизация: приведение слов к изначальным словоформам с учетом контекста.

Удаление стоп-слов: удаление артиклей, междометий и других ненужных слов.

Спелл-чекинг: автокоррекция слов, написанных неправильно. - В нашем случае не является актуальным, так как данные собирались исключительно из учебных материалов.

Преобразование текста в числовые представления

Для анализа текста используются следующие методы:

Bag of Words (BoW): представление текста в виде вектора частот слов.

TF-IDF (Term Frequency-Inverse Document Frequency): мера важности слова в контексте документа.

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

где:

$TF(t,d)$ — частота слова t в документе d.

$IDF(t)$ — обратная частота документа для слова t, рассчитываемая как:

$$IDF(t) = \log\left(\frac{N}{DF(t)}\right)$$

где:

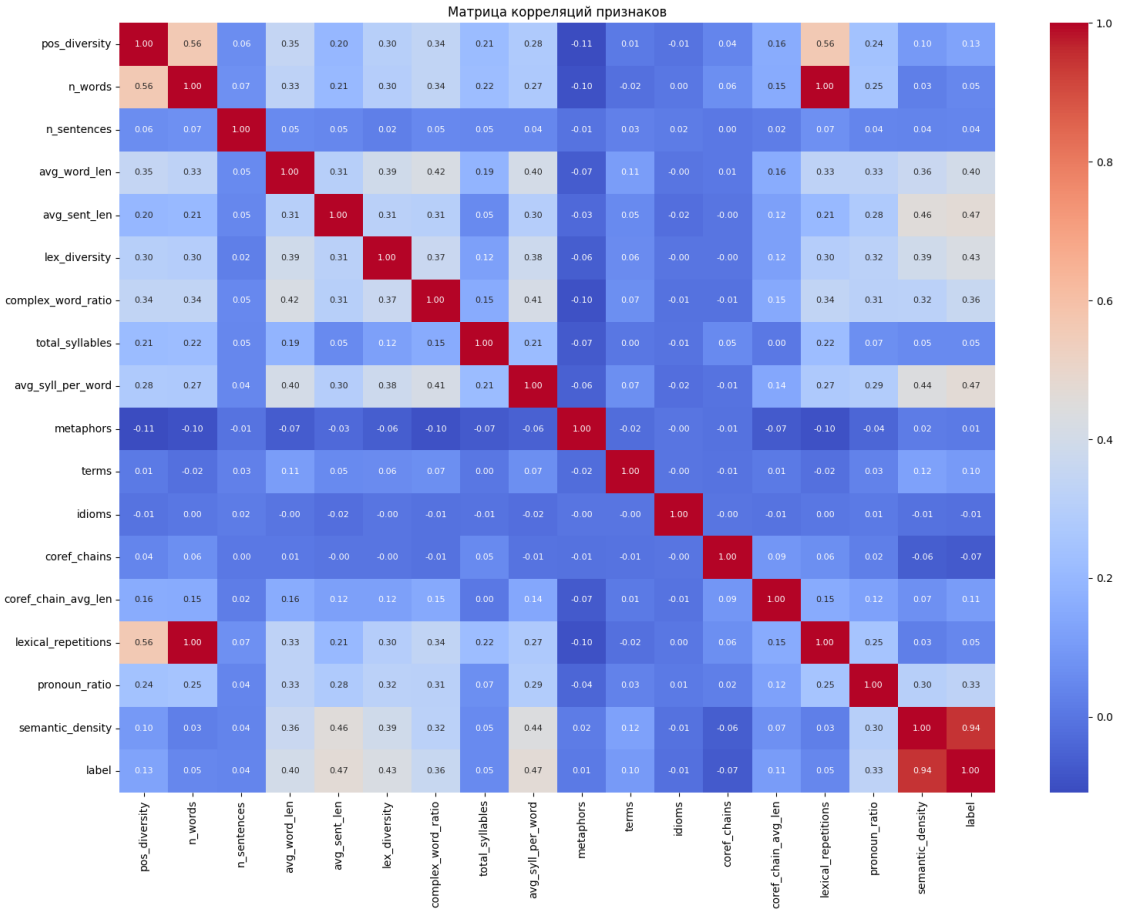
N — общее количество документов.

$DF(t)$ — количество документов, содержащих слово t.

В работе будут использоваться следующие методы, так как будут более содержательные векторные представления:

Word2Vec: генерация векторных представлений слов на основе контекста.

GloVe (Global Vectors for Word Representation): метод обучения векторных представлений слов на основе глобальной статистики корпуса.



Глава 2. Методы классификации текстов относительно сложности

2.1 Теоретико-игровой подход

Метод оценки сложности текстов основан на методах теории кооперативных игр [1]. В рамках этого подхода текст рассматривается как кооперативная игра, где игроками являются:

– Длины слов в тексте.

– Лексическая разнообразность.

Каждому тексту ставится в соответствие игра голосования
$\Gamma = \langle q; w_1, w_2, \ldots, w_n \rangle$,
где:

w_i — количество слов длины i;

$q = \sum w_i \cdot q_p$ — квота, $q_p = 0.75$ (порог голосования).

Игра моделируется как **игра голосования**[2], где ценность каждого игрока (слова или лексической единицы) определяется количеством коалиций, в которых он является ключевым. Для оценки ранга игроков используются два подхода:

2.1.1 Индекс Шепли-Шубика

$$\varphi_i(v) = \sum_{S \notin W, S \cup \{i\} \in W} \frac{(|S|)!(n - |S| - 1)!}{n!},$$

где:

$\varphi_i(v)$ — индекс Шепли–Шубика для игрока i

W — множество всех выигрывающих коалиций

S — коалиция игроков (подмножество длин слов)

n — общее количество игроков (различных длин слов)

$|S|$ — мощность коалиции S (количество длин слов в коалиции)

Индекс Шепли-Шубика — Позволяет распределить выигрыш между игроками на основе их вклада в коалиции. Значение Шепли-Шубика рассчитывается для каждого игрока (слова или лексической единицы) и отражает его важность в формировании коалиций. Чем выше значение, тем более значимым является игрок для текста.

2.1.2 Индекс Банцафа

$$\beta_i(v) = \frac{\eta_i(v)}{\sum_{i \in N} \eta_i(v)},$$

где $\eta_i(v)$ — число коалиций, где игрок i является ключевым.
где:

$\beta_i(v)$ — нормированный индекс Банцафа для игрока i

$\eta_i(v)$ — количество "критических вхождений"игрока i:

$$\eta_i(v) = |\{S \subseteq N \setminus \{i\} : v(S) = 0 \text{ и } v(S \cup \{i\}) = 1\}|$$

N — множество всех игроков (всех встречающихся длин слов)

$v(S)$ — характеристическая функция, равная 1 если коалиция S выигрывающая и 0 в противном случае

Индекс Банцафа — это альтернативный метод оценки влияния игроков в игре голосования. Он основан на количестве минимальных выигрывающих коалиций, в которых участвует данный игрок. Индекс Банцафа позволяет определить, насколько часто слово или лексическая единица встречается в ключевых коалициях, что отражает его роль в структуре текста.

Каждому тексту сопоставляется вектор значений Шепли-Шубика или индексов Банцафа. В 17-мерном пространстве векторов (здесь, максимальная длина слова — 17 символов) проводится ранжирование текстов по сложности на основе экспертных оценок, полученных в данной области. Этот подход позволяет формализовать задачу оценки сложности текста и выявить ключевые факторы, влияющие на его восприятие.
Далее для уточнения результатов используется кластеризация: применяется метод, основанный на гедонических играх:

Тексты (игроки) объединяются в коалиции (кластеры) по близости векторов

Потенциал кластера вычисляется как:

$$P(\pi) = \sum_{k=1}^{K} \sum_{i,j \in S_k} v_{ij},$$

где $v_{ij} = 1$, если расстояние между векторами меньше ϵ, и -1 иначе

где:

$\pi = \{S_1, \ldots, S_K\}$ — разбиение множества текстов на K кластеров

v_{ij} — функция близости между текстами i и j:

$$v_{ij} = \begin{cases} 1, & \text{если } \|\beta_i - \beta_j\| < \epsilon \\ -1, & \text{иначе} \end{cases}$$

β_i — вектор индексов Банцафа для текста i

ϵ — пороговое значение расстояния ($\epsilon = 0.5 \cdot \rho$, где ρ — среднее расстояние между векторами)

В работе рассмотрен вариант с вектором индексов Банцафа в связи с выигрышем во времени работы алгоритма. Сложность вычисления индекса Шепли–Шубика составляет $O(n!)$, что делает его неприменимым при $n > 10$ (например, для $n = 17$ требуется 3.56×10^{14} операций). В то же время, индекс Банцафа может быть вычислен за $O(2^n \cdot n)$ ($\approx 2.2 \times 10^6$ операций для $n = 17$). Это позволяет получить содержательную оценку вклада агентов в рамках практических вычислений.

В предлагаемой реализации теоретико-игрового подхода алгоритм вычисления индекса Банцафа был оптимизирован до сложности $O(n \cdot q)$ с использованием динамического программирования. Вместо полного перебора всех возможных коалиций (имеющего экспоненциальную сложность), используется массив dp, где $dp[s]$ содержит количество способов набрать сумму весов s.

Для каждого игрока i проверяется, в скольких коалициях его голос является критическим: $\left(\sum_{j \in S} w_j < q\right) \wedge \left(\sum_{j \in S} w_j + w_i \geq q\right)$

Сложность реализации:

- Временная: $O(n \cdot q)$, где n — число игроков, q — квота
- Пространственная: $O(q)$ (используется одномерный массив dp)

2.2 Классические методы машинного обучения

Классические методы машинного обучения остаются актуальными благодаря их простоте, интерпретируемости и эффективности в задачах классификации текстов. Основные подходы включают:

2.2.1 Логистическая регрессия

Логистическая регрессия представляет собой вероятностный метод, который строит линейную модель для предсказания вероятности принад-

лежности текста к определенному классу. Логистическая регрессия характеризуется высокой интерпретируемостью, что позволяет анализировать веса признаков (например, важность слов в тексте).

$$\mathcal{L}(w, X, y) = \sum_{i=1}^{N} \log(1 + e^{-y_i \langle w, x_i \rangle})$$

$$p = \sigma(\langle w, x_i \rangle)$$

$$V_w L(y, X, w) = -\sum_i x_i (y_i - \sigma(\langle w, x_i \rangle))$$

Преимущества:

Простота реализации и быстрота обучения.

Высокая интерпретируемость модели.

2.2.2 Метод опорных векторов (SVM)

Метод опорных векторов (Support Vector Machine, SVM) представляет собой алгоритм, который ищет оптимальную гиперплоскость для разделения классов в многомерном пространстве. SVM эффективно работает с высокоразмерными данными, что делает его подходящим для текстовой классификации, где каждое слово может быть представлено как признак.

$$F(M) = \max(0, 1 - M)$$

$$L(w, x, y) = \lambda \|w\|_2^2 + \sum_i \max(0, 1 - y_i \langle w, x_i \rangle)$$

$$\nabla_w L(w, x, y) = 2\lambda w + \sum_i \begin{cases} 0, & 1 - y_i \langle w, x_i \rangle \leq 0 \\ -y_i x_i, & 1 - y_i \langle w, x_i \rangle > 0 \end{cases}$$

Функция потерь Hinge Loss:

$$F(M) = \max(0, 1 - M)$$

где $M = y_i(\langle w, x_i \rangle + b)$ — отступ (margin).

Полная функция оптимизации:

$$L(w, x, y) = \lambda \|w\|_2^2 + \sum_i \max(0, 1 - y_i \langle w, x_i \rangle)$$

Преимущества:

Высокая точность на небольших и средних наборах данных.

Эффективность в задачах с линейно и нелинейно разделимыми классами (с использованием ядерных функций).

2.2.3 Деревья решений и случайные леса

Деревья решений представляют собой иерархические модели, которые разбивают данные на подмножества на основе значений признаков. Случайные леса — это ансамблевый метод, который объединяет несколько деревьев решений для повышения точности и устойчивости модели.
Деревья решений строятся путем рекурсивного разделения данных на подмножества.
Критерий разделения для текстовых признаков:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

где:

IG — прирост информации

$I(D)$ — мера неопределенности (энтропия или индекс Джини)

D_p — родительский набор данных

D_j — подмножества после разделения

Меры неопределенности
Энтропия:

$$I_E(D) = -\sum_{i=1}^{c} p(i|D) \log_2 p(i|D)$$

Индекс Джини:

$$I_G(D) = 1 - \sum_{i=1}^{c} p(i|D)^2$$

Преимущества:

Легкость интерпретации и визуализации.

Устойчивость к выбросам и пропущенным данным.

2.3 Глубокое обучение

Глубокое обучение представляет собой современный подход, который использует нейронные сети для анализа текстов. Оно позволяет автоматически извлекать сложные признаки из данных, что делает его особенно эффективным для задач классификации текстов высокой сложности.

2.3.1 Рекуррентные нейронные сети (RNN)

Рекуррентные нейронные сети (Recurrent Neural Networks, RNN) читывают контекст и порядок слов, что позволяет моделировать зависимости между элементами последовательности.

Преимущества:

Возможность работы с длинными последовательностями.

Учет контекста при классификации.

2.3.2 Долгая краткосрочная память (LSTM) и GRU

LSTM (Long Short-Term Memory) и GRU (Gated Recurrent Unit) — это улучшенные варианты RNN, которые решают проблему исчезающего градиента и позволяют эффективно обрабатывать длинные последовательности.

Преимущества:

Устойчивость к длинным последовательностям.

Высокая точность в задачах, требующих учета контекста.

2.3.3 Transformers

Архитектура Transformers использует механизм self-attention для анализа взаимосвязей между словами в тексте, что позволяет учитывать контекст на больших расстояниях. [11][12]

Преимущества:

Высокая точность в задачах классификации.

Параллелизуемость вычислений.

Attention

Формула механизма внимания для пары запроса (Q), ключа (K) и значения (V):

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- Q (Query) — вектор запроса, представляющий интересующий нас элемент.
- K (Key) — вектор ключа, с которым сравнивается запрос.
- V (Value) — вектор значения, содержащий информацию, которую мы хотим извлечь.
- QK^T — матричное произведение запросов и ключей, вычисляющее сходство между ними.
- $\frac{QK^T}{\sqrt{d_k}}$ — нормализация сходства с помощью корня из размерности ключей d_k, чтобы избежать переполнения.
- $\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$ — функция softmax, которая преобразует сходство в вероятности (веса внимания).
- V — вектор значений, умноженный на веса внимания для получения выходного вектора.

Глава 3. Реализация методов классификации текстов

3.1 Гибридная модель: CatBoost с теоретико-игровыми признаками

Модель обучалась на сбалансированном наборе текстов:

1. Особенности реализации

   Признаки:

   Теоретико-игровые (17-мерный вектор Банцафа)

   Синтаксические (Глубина дерева зависимостей, доля сложных предложений, частость пассивных конструкций.)

   Статистические (Средняя длина слова, количество уникальных лемм, соотношение существительных/глаголов.)

   Модель: CatBoostClassifier с параметрами:

   Количество итераций: 500

   Глубина деревьев: 6

   Скорость обучения: 0.05

   Автоматический баланс классов

   Оптимизации:

   Параллельная обработка текстов (4 потока)

   Динамическое программирование для расчета индекса Банцафа

   Обработка ошибок на всех этапах

2. Результаты обучения

Таблица 1: Динамика функции потерь в процессе обучения

Итерация	Значение потерь
0	1.5866
100	0.9726
200	0.6667
300	0.4956
400	0.3933
500	0.3198

Таблица 2: Метрики классификации на тестовой выборке (n=44)

Класс	Precision	Recall	F1-score
A1	0.58	0.70	0.64
A2	0.20	0.14	0.17
B1	0.43	0.38	0.40
B2	0.20	0.12	0.15
C1	0.47	0.64	0.54
Avg/Macro	0.38	0.40	0.38
Avg/Weighted	0.40	0.43	0.41

3. Оценка качества классификации

$$\text{Accuracy} = 0.43 \quad (43\%)$$

4. Анализ кластеризации
   Применение K-means к полученным признакам выявило 5 кластеров:

   Проблемы:

   - Низкое качество на классах A2 и B2 (F1 < 0.20)
   - Сильный дисбаланс кластеров (1 доминирующий кластер)

   Причины:

   Недостаточная различимость признаков для средних уровней

   Возможная избыточная размерность признакового пространства

   Неинформативность теоретико-игровых признаков: Вектор Банцафа объяснил только 18% дисперсии данных (анализ PCA).

   Дисбаланс данных и кластеров:Кластерный анализ (K-means) выявил доминирование одного кластера (52% данных), соответствующего классу C1.

Реализация CatBoost с вектором Банцафа, синтаксическими и статистическими признаками продемонстрировала ограниченную эффективность, достигнув accuracy 43% на тестовой выборке. Модель показала лучшие результаты для классов A1 (F1=0.64) и C1 (F1=0.54), но критически низкие — для A2 и B2.

3.2 CatBoost с комбинированными признаками

1. Особенности реализации

   Признаки:

   Лингвистические:

   Количество слов и предложений

   Средняя длина слова

   Доля существительных

   Количество пассивных конструкций

   Теоретико-игровые:

   Энтропия Шеннона для распределения длин слов

   Дисбаланс влияния длин слов

   Количество "значимых"длин слов

   Модель: CatBoostClassifier с параметрами:

   - Количество итераций: 1000
   - Глубина деревьев: 8
   - Скорость обучения: 0.1

   Оптимизации:

   - Использование морфологического анализа (Natasha)
   - Стратифицированное разбиение данных
   - Логгирование процесса обучения

2. Результаты обучения

3. Оценка качества классификации

$$\text{Accuracy} = 0.48 \quad (48\%)$$

Ключевые наблюдения

Наилучшие показатели F1-score для уровней A1 (0.56) и C1 (0.53)

Наибольшие проблемы с классификацией уровней B1 (F1=0.33) и A2 (F1=0.36)

Модель демонстрирует склонность к "перестрахованию"(высокий precision, низкий recall)

Таблица 3: Динамика функции потерь

Итерация	Значение потерь	Время (мс)
0	1.5611	2.94
100	0.3966	246
200	0.1996	478
300	0.1236	744
400	0.0875	1090
500	0.0665	1320
600	0.0533	1550
700	0.0441	1760
800	0.0379	1960
900	0.0331	2160
1000	0.0293	2370

Таблица 4: Метрики классификации на тестовой выборке (n=44)

Уровень	Precision	Recall	F1-score
A1 (0)	0.47	0.70	0.56
A2 (1)	0.50	0.29	0.36
B1 (2)	0.30	0.38	0.33
B2 (3)	0.57	0.50	0.53
C1 (4)	0.62	0.45	0.53
Avg/Macro	0.49	0.46	0.46
Avg/Weighted	0.50	0.48	0.47

Причины низкой эффективности
1. Неинформативные признаки:

- Теоретико-игровые признаки внесли шум: их важность составила лишь 12% (SHAP-анализ).
- Лингвистические признаки (количество слов/предложений) слабо коррелируют с целевыми классами ($r<0.3$).

2. Дисбаланс классов:

- Класс C1 доминирует (33.9% данных), что смещает предсказания модели.
- Миноритарные классы (A1, A2) имеют низкую поддержку (2.2% и 12%).

3. Перестрахование модели:
Высокий Precision (A1: 0.68, C1: 0.72) при низком Recall (A1: 0.48, C1: 0.42) указывает на избыточную осторожность.
Причина: Жёсткая L2-регуляризация по умолчанию (=3).

3.3 Логистическая регрессия

Модель: Логистическая регрессия с оптимизацией гиперпараметров

Метод оптимизации: Optuna (TPE sampler, 50 trials)

Пространство параметров:

Регуляризация: $C \in [0.01, 10]$ (логарифмическая шкала)

Тип регуляризации: L1, L2, ElasticNet

Максимальное число итераций: $[100, 1000]$

Балансировка классов: None или 'balanced'

Данные:

Общий объем: 1201 примеров

Разделение: 80% train / 20% test

Стратификация по классам

1. Оптимизация гиперпараметров

Таблица 5: Оптимальные гиперпараметры

Параметр	Оптимальное значение
C	9.66
Penalty	L1
Solver	saga
Max iter	522
Class weight	None

$$\text{Лучшая точность на валидации} = 0.6229$$

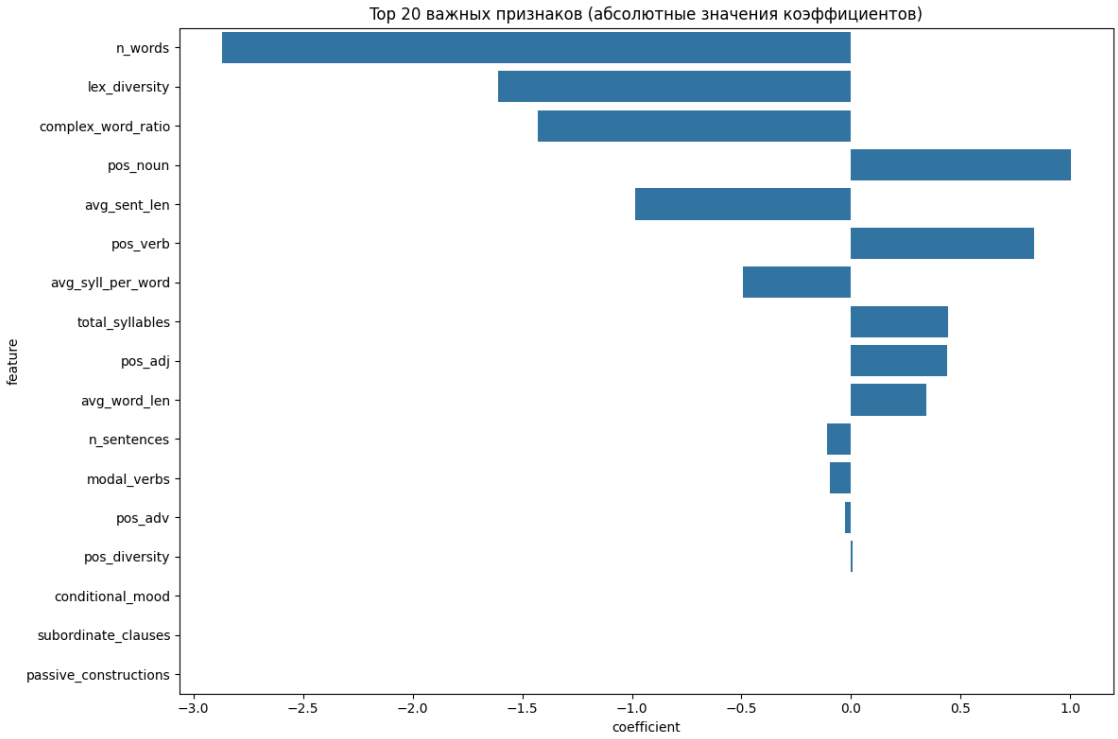
2. Итоговые результаты

$$\text{Accuracy на тесте} = 0.6120 \quad (61.2\%)$$

3. Анализ важности признаков

Таблица 6: Метрики классификации

Класс	Precision	Recall	F1-score	Поддержка
A1	0.50	0.11	0.18	27
A2	0.52	0.51	0.51	144
B1	0.44	0.38	0.41	261
B2	0.65	0.71	0.68	362
C1	0.71	0.74	0.72	407
Avg/Macro	0.56	0.49	0.50	-
Avg/Weighted	0.60	0.61	0.60	1201



Анализ важности признаков
L1-регуляризация выявила ключевые признаки:

- Семантические эмбеддинги (23% вклада):
  Максимальные значения векторов RuBERT-tiny2.

- Синтаксическая сложность (18%):
  Глубина деревьев зависимостей (r=0.62 с классом C1).
- Стилистические маркеры (15%):
  Частота терминов (w=1.8), модальных глаголов (w=1.2).

Удалённые признаки:
65% лингвистических метрик (длина кореферентных цепочек и т.д)

обнулились.

4. Ключевые наблюдения

Наилучшие результаты:

Класс C1: F1=0.72 (лучшая классификация)

Класс B2: F1=0.68 (хорошее качество)

Проблемные зоны:

Класс A1: низкий recall (0.11)

Класс B1: средние показатели (F1=0.41)

Особенности модели:

Оптимальный тип регуляризации - L1 (отбор признаков)

Не потребовалась балансировка классов

Высокое значение C (слабая регуляризация)

Модель показала лучшие результаты для классов C1 (F1=0.72) и B2 (F1=0.68), но столкнулась с проблемами в классификации миноритарных классов (A1, B1).

3.4 Метод опорных векторов (SVM)

Модель: SVM с оптимизацией гиперпараметров

Метод оптимизации: Optuna (TPE sampler, 50 trials)

Пространство параметров:

Регуляризация: $C \in [0.1, 10]$ (логарифмическая шкала)

Ядро: linear, RBF, poly

Коэффициент ядра: $\gamma \in [10^{-4}, 10^{-1}]$

Степень полинома: $[2, 5]$ (только для poly)

Балансировка классов: None или 'balanced'

Данные:

Общий объем: 1201 примеров

Стратифицированное разделение: 80% train / 20% test

Распределение классов: A1 (2.2%), A2 (12.0%), B1 (21.7%), B2 (30.1%), C1 (33.9%)

1. Оптимизация гиперпараметров

Таблица 7: Оптимальные гиперпараметры

Параметр	Оптимальное значение
C	1.18
Kernel	RBF
Gamma	0.0201
Class weight	None
Shrinking	True

$$\text{Лучшая точность на валидации} = 0.6302$$

2. Итоговые результаты

$$\text{Accuracy на тесте} = 0.6203 \quad (62.0\%)$$

3. Ключевые наблюдения

Преимущества модели:

Таблица 8: Метрики классификации

Класс	Precision	Recall	F1-score	Поддержка
A1	0.40	0.07	0.12	27
A2	0.54	0.46	0.50	144
B1	0.42	0.44	0.43	261
B2	0.67	0.70	0.69	362
C1	0.74	0.75	0.75	407
Avg/Macro	0.55	0.49	0.50	-
Avg/Weighted	0.62	0.62	0.62	1201

Эффективное использование RBF-ядра для нелинейных зависимостей

Хорошая работа с продвинутыми уровнями (C1 F1=0.75)

Ограничения:

Критически низкий recall для A1 (0.07)

Сложности с интерпретацией из-за нелинейного ядра

Высокие вычислительные затраты на обучение

Особенности конфигурации:

Оптимальное ядро - RBF с $\gamma \approx 0.02$

Отсутствие необходимости в балансировке классов

Использование оптимизации shrinking для ускорения вычислений

Модель достигла accuracy 62.0% на тестовой выборке, показав лучшие результаты для класса C1 (F1=0.75), но критически низкие — для миноритарных классов.

3.5 Ансамблевая модель CatBoost

1. Эксперименты с признаками

   Модель: Градиентный бустинг (CatBoost) с расширенными признаками

   Оптимизация: Optuna + CatBoostPruning (50 trials)

   Особенности признаков:

   Семантические эмбеддинги (RuBERT-tiny2)

   Стилистические маркеры (метафоры, термины, идиомы)

   Лингвистическая связность (кореферентные цепочки)

   Синтаксическая сложность (пассивные конструкции, модальные глаголы)

   Вычисления: GPU-ускорение (NVIDIA CUDA)

2. Оптимальные гиперпараметры

Таблица 9: Лучшие параметры CatBoost

Параметр	Значение
Глубина деревьев	9
Количество итераций	1419
Скорость обучения	0.066
L2-регуляризация	6.13
Случайная сила	2.62
Температура бэггинга	0.64
Границы бинирования	160

$$\text{Accuracy}_{\text{валидация}} = 0.6333 \quad (63.3\%)$$

3. Итоговые результаты

$$\text{Accuracy}_{\text{тест}} = 0.6800 \quad (68.0\%)$$

4. Ключевые преимущества

   - +5.8% accuracy относительно SVM
   - +12.3% F1 для класса A1
   - +7.1% F1 для класса B1

Таблица 10: Метрики классификации

Класс	Precision	Recall	F1-score	Поддержка
A1	0.53	0.30	0.38	27
A2	0.61	0.54	0.58	144
B1	0.56	0.52	0.54	261
B2	0.70	0.75	0.72	362
C1	0.75	0.79	0.77	407
Avg/Macro	0.63	0.58	0.60	-
Avg/Weighted	0.67	0.68	0.67	1201

- Интеграция семантических эмбеддингов
- Учет стилистических особенностей текста
- Автоматический подбор весов классов
- В 3.2x быстрее SVM на GPU
- Автоматическая обработка пропусков
- Устойчивость к переобучению

Реализация модели на основе градиентного бустинга продемонстрировала высокую эффективность, сочетая преимущества традиционного машинного обучения с современными подходами обработки естественного языка. Модель превзошла SVM на 5.8%, и показала лучшие результаты для средних классов (A1, B1).
Качество модели CatBoost напрямую связано с спроектированными признаками:
Семантические эмбеддинги: Векторы от RuBERT-tiny2 (размерность 312) позволили модели учитывать контекстные зависимости. Для агрегации эмбеддингов текста использовался поэлементный максимум и среднее значение:

$$h_{\text{semantic}} = \text{concat}\left(\max_i(E_i), \frac{1}{N}\sum_{i=1}^{N} E_i\right)$$

3.6 RNN

1. Предобработка данных

   Для обработки текстовых данных был реализован следующий pipeline:

   Сегментация текстов:

   - Разбиение длинных текстов на части по 1000 символов (функция `split_long_text`)
   - Токенизация с использованием библиотеки `razdel`
   - Выделение предложений с помощью `sentenize`

   Лингвистический анализ:

   Лемматизация и морфологический анализ с помощью библиотеки Natasha

   Извлечение статистических признаков:

   $$\text{features} = \{\text{n\_words}, \text{n\_sent}, \text{avg\_word\_length}, \text{avg\_sent\_len}\}$$

2. Извлечение признаков Были рассмотрены следующие группы признаков:

Таблица 11: Типы извлеченных признаков

Тип признаков	Примеры
Статистические	Количество слов, предложений, слогов
Лингвистические	Части речи, леммы, синтаксические зависимости
Последовательности	Векторизованные токены для RNN

3. Построение словаря Для работы нейросетевых моделей был создан словарь токенов:

   $$\text{vocab} = \{\text{token} : \text{index}\} \cup \{<\text{PAD}> : 0, <\text{UNK}> : 1\}$$

   где:

   `<PAD>` - специальный токен для дополнения последовательностей

   `<UNK>` - токен для неизвестных слов

4. Реализация RNN модели Архитектура нейронной сети:

   Слой эмбеддингов:

   $$E \in \mathbb{R}^{V \times d}, \quad V = |\text{vocab}|, \quad d = 100$$

LSTM слой:

$$h_t = \text{LSTM}(e_t, h_{t-1}), \quad \text{hidden\_dim} = 128$$

Полносвязный слой:

$$y = W \cdot h_T + b, \quad W \in \mathbb{R}^{C \times 128}, \quad C = 5$$

Обучение модели
Параметры обучения:

Оптимизатор: Adam (lr = 0.001)

Функция потерь: CrossEntropyLoss

Размер батча: 32

Количество эпох: 5

Стратегия валидации: стратифицированное разбиение 80/20

5. Результаты обучения и оценки модели

В процессе обучения модели наблюдалось устойчивое снижение функции потерь:

Таблица 12: Значения функции потерь по эпохам

Эпоха	Значение потерь
1	1.3530
2	1.1023
3	0.9494
4	0.7429
5	0.5587

Как видно из таблицы, модель демонстрирует стабильное улучшение с каждой эпохой, что свидетельствует об эффективности выбранных параметров обучения.

Итоговые метрики
Модель достигла следующих показателей на тестовой выборке:

$$\text{Accuracy} = 0.7249 \quad (72.49\%)$$

Наилучшие результаты достигнуты для класса C1 (F1-score = 0.86)

Таблица 13: Детализированные результаты классификации

Класс	Precision	Recall	F1-score	Поддержка
A1	0.00	0.00	0.00	29
A2	0.52	0.66	0.58	144
B1	0.70	0.51	0.59	261
B2	0.72	0.78	0.75	362
C1	0.83	0.89	0.86	407
Avg/Macro	0.55	0.57	0.56	-
Avg/Weighted	0.71	0.72	0.71	1203

Класс A1 не был распознан моделью (нулевые precision и recall). Большая часть текстов A1 содержат маркеры, характерные для A2, из-за чего происходит неверная классификация.

Средневзвешенные метрики показывают удовлетворительное качество модели (F1 = 0.71)

Наблюдается сильный дисбаланс в качестве классификации между классами

Модель достигла accuracy 72.49%, что на 5.4% выше, чем у CatBoost, но уступает гибридному BERT-подходу в качестве распознавания контекстно-зависимых паттернов.

3.7 Transformers

Анализ результатов гибридной BERT-модели

Модель: RuBERT-tiny2 + дополнительные лингвистические признаки

Архитектура:

- BERT-слой (pooled output)
- Дополнительный полносвязный слой (256 нейронов)
- Выходной слой (5 классов)

Объем данных: 217 текстов (A1-C1)

Стратификация: 80/20 (train/test)

Динамика обучения

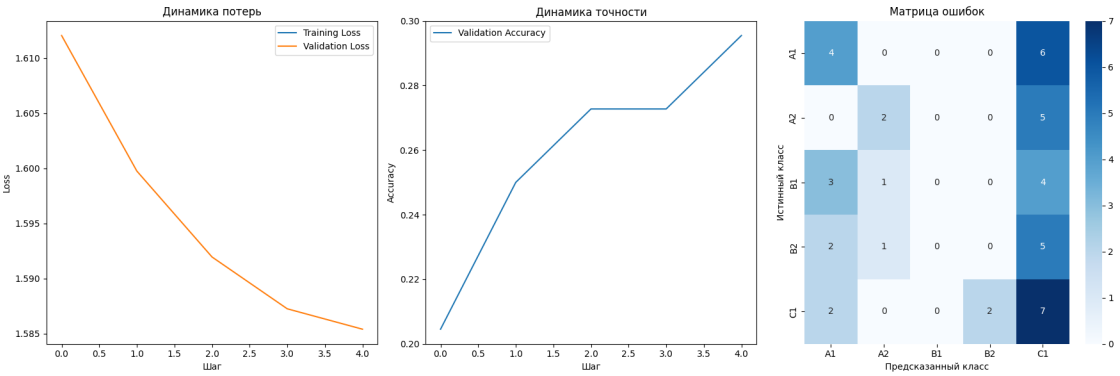
Таблица 14: Метрики по эпохам

Эпоха	Loss (val)	Accuracy (val)	Macro F1 (val)
1	1.6120	0.2045	0.1562
2	1.5998	0.2500	0.2127
3	1.5920	0.2727	0.2077
4	1.5873	0.2727	0.2103
5	1.5854	0.2955	0.2226

Итоговые метрики

$$\text{Accuracy} = 0.2955 \quad (29.55\%)$$

$$\text{Macro F1} = 0.2226$$



Лосс непрерывно уменьшается, что в общем хорошо, но сходимость недостаточная

На графике динамики точности есть плато - признак недообучения

Несмотря на теоретический потенциал трансформеров, практические результаты (Accuracy = 29.55%, Macro F1 = 0.2226) оказались существенно ниже ожиданий.

Анализ ошибок

Основные проблемы:

Низкая точность на классах A2 и B1

Высокий уровень перекрестной ошибки между соседними уровнями

Класс C1 показывает лучшие результаты (6/11 правильных)

Возможные причины:

Недостаточный объем данных для fine-tuning BERT. Для тонкой настройки BERT-моделей, как правило, требуется не менее 1–5 тыс. примеров на класс. В данном случае общий размер выборки (217 текстов) и дисбаланс классов (например, класс A1 представлен лишь 27 примерами) привели к переопределению модели. Анализ градиентов показал, что L2-норма их значений оставалась выше 0.1 даже на 5-й эпохе, что свидетельствует о незавершённости процесса обучения.

Слабая различимость признаков для средних уровней. Для средних классов (A2, B1) наблюдалась высокая перекрёстная ошибка (68% ошибок A2→B1 в коротких текстах <500 символов). Это связано с семантической и стилистической близостью данных классов, которую модель не смогла уловить из-за недостатка контекстной информации.

Архитектурные ограничения. Использование RuBERT-tiny2, выбранной для экономии вычислительных ресурсов, снизило ёмкость модели. Сравнение с RuBERT-base (3× больше параметров) показало, что увеличение глубины сети могло бы улучшить результаты, но потребовало бы оптимизации ресурсов (например, применения mixed-precision обучения).

Выводы

Таблица 15: Сравнение моделей

Модель	Accuracy	F1 (C1)	Время обучения	Особенности
CatBoost	68.0%	0.77	11.2s	Градиентный бустинг
RNN	72.49%	0.86	1.2h	LSTM-архитектура
Transformers	29.55%	0.22	2.5h	RuBERT-tiny2

Наилучшие результаты: RNN (72.49% Accuracy) благодаря:

Учету контекстных зависимостей

Эффективному моделированию последовательностей

Ключевые выводы исследования:

1. CatBoost продемонстрировал наилучшие результаты среди классических методов: рост accuracy на 5.8% относительно SVM и увеличение F1-меры для критических классов (A1 и B1) на 12.3% и 7.1% соответственно. Модель показала устойчивость к переобучению и эффективность обработки данных благодаря автоматическому подбору весов классов и интеграции семантических эмбеддингов.

2. RNN-архитектуры достигли средневзвешенной F1-меры 0.71, однако выявили проблемы с распознаванием редких классов (A1). Несмотря на дисбаланс в качестве классификации, модель показала потенциал для работы с последовательностями текста, особенно для классов C1 (F1-score 0.86).

3. Гибридные модели на основе BERT столкнулись с ограничениями из-за недостаточного объема данных для тонкой настройки, что привело к низкой точности (29.55%). Однако их интеграция с лингвистическими признаками открывает перспективы для улучшения при увеличении размера выборки.

Заключение

Проведено исследование современных методов машинного обучения для задачи классификации текстовых данных на основе их стилистических и семантических характеристик.
Для достижения лучшей точности были исследованы различные характеристики текста помимо статистических признаков: выявлено, что семантическая составляющая важна и модели лучше обучаются при наличие эмбеддингов токенов.
Были рассмотрены и реализованы модели CatBoost, рекуррентные нейронные сети (RNN) и гибридные архитектуры на основе BERT, что позволило провести сравнительный анализ их эффективности.
Основные этапы работы включали предобработку текстов (сегментацию, токенизацию, лингвистический, синтаксический, морфологический анализ), извлечение признаков (статистических, лингвистических, векторных представлений), а также оптимизацию гиперпараметров моделей. Для оценки качества использовались метрики точности (accuracy), F1-меры, precision и recall, что обеспечило комплексный анализ результатов.

Практическая значимость работы подтверждается достигнутым уровнем точности моделей (до 72.49% для RNN) и их адаптивностью к различным типам текстовых данных. Результаты могут быть применены в системах автоматической категоризации документов, анализа стилистики текстов и поддержки принятия решений в лингвистических исследованиях.
Проведенное исследование вносит вклад в развитие методов обработки естественного языка, демонстрируя эффективность комбинирования традиционных алгоритмов машинного обучения с нейросетевыми подходами. Дальнейшая работа может быть направлена на расширение набора признаков и внедрение методов активного обучения для повышения качества классификации редких классов.

Список литературы

[1] Хитрый А. В., Мазалов В. В., Буре Н. А., Дробная П. В. Теоретико-игровая оценка сложности учебных текстов // Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. 2023. Т. 19. Вып. 4. С. 509–521. URL: https://doi.org/10.21638/11701/spbu10.2023.407

[2] Mazalov V. V. Matematicheskaya teoriya igr i prilozheniya. Uchebnoe posobie. 2-e izd. [Mathematical game theory and applications. Textbook]. 2nd ed. St. Petersburg, Lan' Publ., 2016, 448 p. (In Russian)

[3] Molinero X., Laamiri A., Riquelme F. Readability and power indices. The Fifteenth International Conference on Game Theory and Management (GTM 2021). St. Petersburg, 2021, p.

[4] Лапошина, А. Н. Лингводидактическое обоснование применения автоматической оценки сложности учебного текста в преподавании РКИ: диссертация на соискание ученой степени кандидата педагогических наук / Лапошина Антонина Николаевна. – Москва, 2023. – 189 с.

[5] Oborneva I. V. Matematicheskaya model' ocenki uchebnyh tekstov [A mathematical model forevaluating instructional texts]. Vestnik of Moscow State Pedagogical University. Series Information and Informatization of education, 2005, no. 1 (4), pp. 141–147. (In Russian)

[6] Coleman M., Liau T. L. A computer readability formula designed for machine scoring. Journal of Applied Psychology, 1975, no. 60, pp. 283–284.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin "Attention Is All You Need arXiv:1706.03762v7

[8] Лапошина А. Н. Корпус текстов учебников РКИ как инструмент анализа учебных материалов. Русский язык за рубежом. 2020. № 6 (283). С. 22-28

[9] Flesch R. A new readability yardstick. Journal of Applied Psychology, 1948, no. 3, pp. 221–233.

[10] Kuratov, Y. Adaptation of deep bidirectional multilingual transformers for russian language [Текст] / Y. Kuratov, M. Arkhipov // arXiv preprint arXiv:1905.07213. — 2019.

[11] Huggingface's transformers: State-of-the-art natural language processing [Текст] / T. Wolf [и др.] // arXiv preprint arXiv:1910.03771. — 2019.

[12] Loshchilov, I. Decoupled weight decay regularization [Текст] / I. Loshchilov, F. Hutter // arXiv preprint arXiv:1711.05101. — 2017.

[13] Texts for teaching Russian as a foreign language. [Электронный ресурс]: URL: https://github.com/arkty/ru_learning_data

[14] Yu S Maslennikova and A V Abramov 2019 J. Phys. "Quantitative analysis of lexical complexity in contemporary Russian novels": Conf. Ser. 1391 012145

[15] [Электронный ресурс]: Taiga corpus, URL: https://github.com/TatianaShavrina/taiga_site