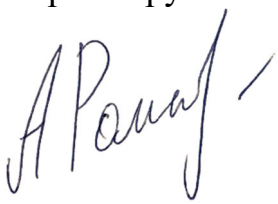
Федеральное государственное автономное образовательное учреждение
высшего образования
«Томский государственный университет систем управления
и радиоэлектроники» (ТУСУР)

На правах рукописи



Романов Александр Сергеевич

МЕТОДОЛОГИЯ ИДЕНТИФИКАЦИИ АВТОРА ТЕКСТОВОЙ ИНФОРМАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ

Специальность 2.3.6 – Методы и системы защиты информации,
информационная безопасность

Диссертация на соискание ученой степени доктора технических наук

Научный консультант
доктор технических наук, профессор
Шелупанов Александр Александрович

Томск – 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 7

1 ОБЗОР ПРОБЛЕМ И РЕШЕНИЙ В ОБЛАСТИ АНАЛИЗА ТЕКСТОВ ДЛЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ 20

1.1 Идентификация автора текста и защита интеллектуальной собственности 20
1.1.1 Методы статистического анализа 23
1.1.2 Методы машинного обучения 25
1.1.3 Методы глубокого обучения 28
1.2 Определение пола и гендера, возраста автора. выявление признаков лгбт и педофилии 32
1.2.1 Определение пола и возраста автора текста 33
1.2.2 Определение признаков ЛГБТ в тексте 38
1.2.3 Модели компьютерного зрения для определения возраста автора 40
1.3 Идентификация автора сообщений в сети интернет и продленная аутентификация пользователя социальных сетей 41
1.4 Определение автора программного кода 47
1.5 Определение настроения и деструктивной направленности в текстах 56
1.5.1 Определение тональности текста 58
1.5.2 Определение деструктивной и экстремистской направленности в текстах 62
1.6 Анонимизация, имитация и генеративные модели как методы атаки на методы идентификации автора 69
1.6.1 Обфускация 71
1.6.2 Имитация 73
1.6.3 Генеративные модели 75
1.6.4 Атаки на методы идентификации автора программного кода 77
1.7 Выводы по главе 80

2 МЕТОДОЛОГИЯ ИДЕНТИФИКАЦИИ АВТОРА ТЕКСТОВОЙ ИНФОРМАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ 83

2.1 Этапы создания методологии 83
2.2 Авторская методология идентификации автора для решения задач кибербезопасности 85
2.3 Авторская модель создания автором текста в киберсреде с учетом семантики 88
2.4 Используемые модели представления текста 99
2.4.1 Модель «мешка слов» (BOW) 99
2.4.2 Модели на основе *n*-грамм 100

2.4.3 Модель на основе TF-IDF 102
2.4.4 Модели на основе Word2Vec 103
2.4.5 Модели на основе предобученных эмбеддингов 105
2.5 Используемые алгоритмы разбора текста 107
2.6 Используемые методы семантического анализа 109
2.6.1 Методы иерархической кластеризации 110
2.6.2 Методы на основе плотности объектов 110
2.6.3 Методы на основе центроидов 111
2.6.4 Методы на основе плотности вероятности 111
2.6.5 Методы на основе графов 112
2.6.6 Методы на основе плотности признаков 113
2.6.7 Методы на основе статистических моделей 114
2.6.8 Семантические нейросетевые методы 114
2.7 Используемые методы принятия решений 116
2.7.1 Методы на основе статистики 117
2.7.2 Методы на основе классических алгоритмов машинного обучения 120
2.7.3 Методы на основе нейронных сетей 124
2.7.4 Методы на основе глубокого обучения 127
2.8 Используемые методы оптимизации гиперпараметров 135
2.9 Используемые методы снижения размерности 137
2.9.1 Методы на основе фильтрации 138
2.9.2 Методы на основе оберточных моделей 141
2.10 Используемые методы сглаживания 143
2.11 Различие естественно- и искусственно-языковых текстов 144
2.12 Выводы по главе 145
3 МЕТОДИКА ИДЕНТИФИКАЦИИ АВТОРА ЕСТЕСТВЕННО-ЯЗЫКОВОГО ТЕКСТА 147
3.1 Закрытая и открытая атрибуция, верификация автора 147
3.2 Методика идентификации автора для случая закрытой атрибуции 149
3.2.1 Предварительная обработка данных 150
3.2.2 Формирование признакового пространства в задаче идентификации автора текста 150
3.2.3 Отбор информативных признаков 151
3.2.4 Валидация модели 152
3.3 Результаты экспериментов по определению авторства в случае закрытого множества кандидатов 155
3.3.1 Художественные тексты 155
3.3.2 Тексты авторов-любителей 164

3.3.3 Короткие тексты пользователей социальных сетей 167
3.3.4 Случай использования генеративных моделей 171
3.4 Методика верификации автора 173
3.5 Методика идентификации автора для случая открытой атрибуции 175
3.5.1 Метод на основе комбинации одноклассового SVM и GRU+CNN 176
3.5.2 Метод на основе распределения вероятностей Softmax при использовании GRU+CNN 176
3.5.3 Метод на основе порогового значения расстояния Евклида 177
3.5.4 Метод на основе порогового значения косинусного сходства 177
3.6 Результаты экспериментов по определению авторства в случае открытого множества кандидатов 178
3.6.1 Художественные тексты 179
3.6.2 Тексты авторов-любителей 180
3.6.3 Короткие тексты пользователей социальных сетей 182
3.7 Оценка разработанных методик в сравнении с аналогами 182
3.8 Выводы по главе 183
4 МЕТОДИКА ИДЕНТИФИКАЦИИ АВТОРА ИСХОДНОГО КОДА ПРОГРАММЫ 186
4.1 Выбор подходов к идентификации автора исходного кода программы 187
4.2 Сложные случаи идентификации автора исходного кода 191
4.2.1 Обфускация исходного кода 191
4.2.2 Использование стандартов кодирования 192
4.2.3 Использование нескольких языков программирования 193
4.2.4 Искусственно-сгенерированные исходные коды 194
4.2.5 Коммиты исходных кодов программ 195
4.3 Предварительная оценка подходов к идентификации автора исходного кода программы 196
4.4 Методика идентификации автора исходного кода программы на основе CodeBERT 197
4.5 Результаты экспериментов 198
4.5.1 Результаты экспериментов для простых случаев идентификации автора исходного кода 200
4.5.2 Результаты экспериментов для сложных случаев идентификации автора исходного кода 201
4.6 Выводы по главе 207
5 ПРИМЕРЫ ПРИМЕНЕНИЯ РАЗРАБОТАННОЙ МЕТОДОЛОГИИ К РЕШЕНИЮ ВАЖНЫХ НАРОДНО-ХОЗЯЙСТВЕННЫХ ЗАДАЧ 209

5.1 Методика определения текстов деструктивной и экстремистской направленности 209
5.1.1 Описание методики 210
5.1.2 Семантическая кластеризация 212
5.1.3 Трансферное обучение 219
5.1.4 Определение автора текста, имеющего деструктивную и экстремистскую направленность 220
5.1.5 Результаты исследования методики 220
5.1.6 Выводы по подразделу 230
5.2 Методика определения возраста автора текстовой информации 232
5.2.1 Описание методики 232
5.2.2 Результаты экспериментов 239
5.2.3 Выводы по подразделу 241
5.3 Методика идентификации пола и гендера автора текстовой информации, включая представителей ЛГБТ-сообщества 242
5.3.1 Описание методики 243
5.3.2 Формирование признакового пространства 244
5.3.3 Ансамбль на основе SVM, CNN и BERT 246
5.3.4 Постановка экспериментов и результаты 247
5.3.5 Оценка методики в сравнении с аналогами 253
5.3.6 Выводы по подразделу 254
5.4 Методика проверки однородности текста и поиска заимствований 255
5.4.1 Описание методики 256
5.4.2 Результаты экспериментов по выявлению плагиата 260
5.4.3 Выводы по подразделу 264
5.5 Выводы по главе 265

6 АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, НАБОРЫ ДАННЫХ ДЛЯ АНАЛИЗА ЕСТЕСТВЕННО- И ИСКУССТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ В ЗАДАЧАХ КИБЕРБЕЗОПАСНОСТИ 266

6.1 Программный комплекс для получения информативных признаков текста на основе семантического анализа «PyAuthorship» 266
6.1.1 Структура программного комплекса 266
6.1.2 Подсистема сбора образцов текстов 267
6.1.3 Подсистема хранения информации 269
6.1.4 Подсистема анализа естественного языка 274
6.1.5 Подсистема анализа искусственного языка 279
6.1.6 Подсистема выделения информативных признаков 279

6.2 Программная система для идентификации автора письменной речи «Автороведъ» 284
6.3 Чат-бот для определения возраста автора текстового сообщения «Age Detector» 287
6.4 Система для идентификации автора исходного кода программы «CoDEtective» 289
6.5 Программа для тонального анализа текста «Moodorrude» 292
6.6 Программный комплекс для идентификации запрещенных законодательством РФ и деструктивных текстов «PyDestructiveDetector» 293
6.7 Базы данных естественно и искусственно-языковых текстов для определения авторства 297
6.7.1 Наборы данных для проведения экспериментов по идентификации автора естественно-языкового текста 297
6.7.2 Набор данных для проведения экспериментов по идентификации автора исходного кода 303
6.7.3 Набор данных для проведения экспериментов по определению возраста 305
6.7.4 Наборы данных для проведения экспериментов по определению деструктивной и экстремисткой направленности текста 306
6.7.5 Набор данных для экспериментов по определению пола и гендера 308
6.7.6 Набор данных для экспериментов по проверке однородности и поиску плагиата 311
6.7.7 Достоверность данных 312
6.8 Выводы по главе 316
ЗАКЛЮЧЕНИЕ 318
СПИСОК ЛИТЕРАТУРЫ 322
ПРИЛОЖЕНИЕ А СРАВНЕНИЕ ОСНОВНЫХ РАБОТ В ОБЛАСТИ АНАЛИЗА ТЕКСТОВ ДЛЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ 356
ПРИЛОЖЕНИЕ Б ПСЕВДОКОДЫ АЛГОРИТМОВ 374
ПРИЛОЖЕНИЕ В СВИДЕТЕЛЬСТВА О РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ И БАЗ ДАННЫХ 377
ПРИЛОЖЕНИЕ Г АКТЫ ВНЕДРЕНИЯ И ИСПОЛЬЗОВАНИЯ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ 388

ВВЕДЕНИЕ

Актуальность темы

В эпоху, когда вычислительные технологии стали неотъемлемой частью всех аспектов человеческой деятельности, критически важным является обеспечение защиты данных и систем от киберугроз. Первостепенной задачей становится разработка и внедрение передовых систем и технологий для обеспечения информационной безопасности. Одним из направлений, где требуются такие инструменты, является анализ больших объемов текстовых данных, направленный на решение специфических задач кибербезопасности, в частности, определения авторства текста.

В контексте информационной безопасности идентификация автора текста играет ключевую роль в борьбе с киберпреступностью. С ее помощью становится возможным выявление и пресечение действий распространителей ложной и вредоносной информации, такой как пропаганда экстремизма, терроризма и движений, противоречащих традиционным ценностям России, а также розжига ненависти и розни. В образовательной сфере тема актуальна для обеспечения объективности оценки и подтверждения подлинности работ. В области лингвистики она предоставляет уникальные возможности для изучения стилистических особенностей языка, разработки новых методов анализа и понимания эволюции языка.

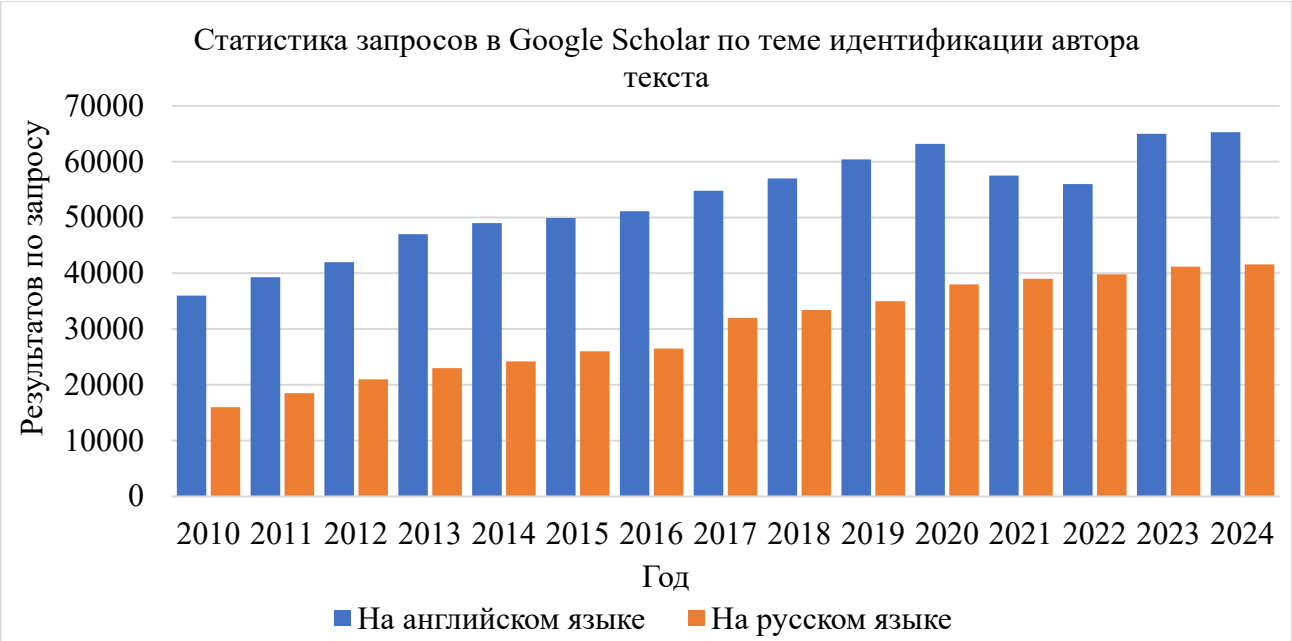


Рисунок – Статистика запросов по теме исследования

Задача идентификации автора текстовой информации не теряет актуальности уже более 100 лет, что подтверждается статистическими данными о публикациях по теме. На рисунке выше представлены данные о количестве результатов, полученных для запросов к базе цитирования Google Scholar на русском и английском языке на тему идентификации автора текста. Результаты охватывают общее количество публикаций, тезисов и статей по теме за период с 2010 (год защиты автором кандидатской диссертации) по 2024 год. Видно, что интерес к теме российских и зарубежных исследователей только растет.

Наибольший вклад в решение отдельных задач определения автора текстовой информации и атрибутов автора в России был внесен Н.А. Морозовым, А.А. Марковым, В.П. Фоменко и Т.Г. Фоменко, Д.В. Хмелевым, Ю.В. Сидоровым, А.Ю. Комиссаровым, О.Г. Шевелевым, В.В. Поддубным, М.А. Марусенко, Р.В. Мещеряковым, А.О. Исхаковой, А.А. Роговым, Н.Д. Москиным, А.Г. Сбоевым, М.Е. Сухопаровым, В.Д. Стремоуховым и др.

Наиболее цитируемыми зарубежными авторами, занимавшимися частными вопросами определения авторства и атрибуции, являются A. Abbasi, T.C. Menden hall, A.Q. Morton, E. Stamatatos, J. Weerasinghe, M. Najafi, H. Gómez Adorno, D. Hernández, S. Burrows, B. Alsulami, W. Wisse, F. Ullah, S. Hedegaard, M. Yang, G. Farnadi, D. Nguyen, C.Y. Park, S. Afroz и др.

Перечисленные исследователи успешно решали частные задачи атрибуции, не связанные с обеспечением кибербезопасности напрямую. Однако проблема систематизации существующих подходов к идентификации автора текста, а также разработки комплексной методологии, позволяющей эффективно решать взаимосвязанные задачи авторства в интересах национальной кибербезопасности страны, остается открытой.

В диссертационной работе предлагается комплексное решение **важной научной проблемы** идентификации автора текстовой информации и таких авторских атрибутов, как пол, возраст, идеология и взгляды, основанное на передовых технологиях искусственного интеллекта и машинного обучения, соответствующих приоритету научно-технологического развития России на 2027–2032 год, согласно

Указу Президента Российской Федерации № 642 от 01.12.2016 г. (ред. от 15.03.2021) «О Стратегии научно-технологического развития Российской Федерации», п. 20, пп. «а».

Цель и задачи исследования

Целью диссертационной работы является создание методологии идентификации автора текстовой информации, включая естественно-языковые тексты и исходные коды программ, для решения задач кибербезопасности.

Для достижения поставленной цели необходимо решить следующие задачи.

1. Провести анализ современного состояния проблемы идентификации автора текстовой информации и использующихся для этого методов анализа и характеристик текста, выделить актуальные задачи кибербезопасности, связанные с определением авторства текста.

2. Разработать обобщенную методологию идентификации автора текстовой информации для решения задач кибербезопасности, учитывающую специфику естественно- и искусственно-языковых текстов.

3. Разработать прикладные методики идентификации автора текстовой информации, сочетающие отдельные элементы методологии.

4. Разработать информационное, алгоритмическое и программное обеспечение для решения задач кибербезопасности, связанных с идентификацией автора текстовой информации.

5. Исследовать и апробировать разработанные методики для решения задач кибербезопасности на корпусах текстов.

Объектом исследования является печатный текст и его характеристики.

Предметом исследования являются характеристики текста, описывающие авторский стиль, методы и алгоритмы машинного обучения, предназначенные для работы с естественно- и искусственно-языковыми текстами.

Методы исследования

Для решения задач, сформулированных в работе, использовались методы математической статистики, вычислительного эксперимента и искусственного

интеллекта. При разработке программной системы использовались методы объектно-ориентированного программирования.

Научная новизна полученных результатов проведенного диссертационного исследования и полученных результатов заключается в следующем:

1. **Впервые** предложена комплексная методология идентификации автора текста, учитывающая особенности естественно- и искусственно-языковых текстов и возможные атаки на методы идентификации.

2. Предложена модель создания текста автором в киберсреде, **впервые** учитывающая семантические особенности и информативные признаки текста на разных уровнях иерархического анализа, специфику среды, атрибуты автора и вид деятельности по созданию текста.

3. Предложена методика идентификации автора естественного текста, **отличающаяся** использованием комбинации GRU+CNN и SVM с отбором признаков генетическим алгоритмом и методом на основе регуляризации, **впервые** учитывающая случаи открытой и закрытой атрибуции и текстов, созданных генеративными нейросетями.

4. Предложена методика идентификации автора исходного кода программ, в которой **впервые** используется классификатор на основе глубокой модели CodeBERT. Методика **отличается** учетом комплекса сложных случаев идентификации (обфускация, командная разработка, использование стандартов кодирования и искусственно-сгенерированных кодов программ).

5. Предложена методика бинарной и мультиклассовой классификации по возрастным группам на основе текста, **отличающаяся** использованием модели fastText в сочетании с методами компьютерного зрения для фильтрации обучающих данных.

6. Предложена методика определения тестов деструктивной и экстремистской направленности и определения автора таких текстов, **отличающаяся** использованием моделей GRU+CNN и BERT, методов семантической кластеризации текста и трансферного обучения, что позволяет выявлять запрещенные законодательством РФ тексты.

7. Разработана **новая** методика определения пола и гендера автора русскоязычного текста, **впервые** учитывающая гендеры ЛГБТ* и отличающаяся использованием ансамбля: SVM, обученного на признаковом пространстве, сформированного на основе методов семантической кластеризации, сверточной сети, обученной на частотных распределениях триграмм, сглаженных методом Катца, и BERT.

8. Предложена методика проверки однородности текста и поиска заимствований, **отличающаяся** применением сиамских НС SimNN, обученных с контрастивной и тройной функциями потерь. Методика **впервые** решает задачу выявления заимствований для случаев открытого множества авторов-кандидатов и использования искусственной генерации текстов.

Достоверность и **обоснованность** научных положений, результатов и основных выводов работы обеспечивается корректностью применяемых методов математической статистики, вычислительного эксперимента и искусственного интеллекта, серией практических экспериментов на представительном и репрезентативном корпусе текстов, а также согласованностью полученных данных с результатами других научных групп и положительным эффектом от внедрения полученных результатов.

Теоретическая значимость работы состоит в проработке и систематизации результатов предыдущих исследований, их дополнении и комплексировании с вновь разработанными методиками, моделями и алгоритмами в составе единой методологии идентификации автора текстовой информации.

Практическая значимость

1. Разработаны репрезентативные корпуса текстовой информации, включающие художественные, любительские тексты и сообщения из социальных сетей и мессенджеров, исходные коды программ, содержащие атрибуты автора (пол, возраст, идеология и др.), которые можно использовать для обучения моделей при решении задач кибербезопасности.

2. Предложенная методика идентификации автора естественно-языкового текста и её программная реализация позволяют существенно сократить

* Запрещенная в Российской Федерации экстремистская организация.

необходимый объем текста для идентификации: до 15000 символов для обычных и до 250 символов для коротких текстов, при этом повышая точность на 2–14% по сравнению с аналогами. Методика учитывает случаи закрытой и открытой атрибуции, устойчива к внедрению текстов, созданных генеративными языковыми моделями на основе образцов текстов реальных авторов.

3. Предложенная методика идентификации автора исходного кода и ее программная реализация позволяют повысить точность идентификации автора на 20–30% по сравнению с существующими подходами для современных интерпретируемых и компилируемых языков программирования в простых и осложненных различными факторами (обфускация, командная разработка, стандарты кодирования, искусственная генерация) случаях.

4. Предложенная методика определения текстов деструктивной и экстремистской направленности и ее программная реализация позволяют получить прирост точности выявления таких текстов в сравнении с аналогами до 32%, а также повысить точность идентификации их автора на 22%, что позволяет выявлять запрещенные законодательством РФ тексты.

5. Предложенная методика определения возраста автора текстовой информации и её программная реализация позволяет определять принадлежность автора к конкретной возрастной группе с точностью на 2% превосходящей аналоги, а также учитывать тексты, написанные несовершеннолетними авторами, что не предусматривается аналогами методики.

6. Предложенная методика идентификации пола и гендера автора текстовой информации и её программная реализация позволяет определять пол с точность на 9% выше, чем аналоги, учитывать гендеры и выявлять тексты, созданные представителями ЛГБТ-сообщества* с точностью 93%, что позволяет выявлять пропаганду нетрадиционных ценностей, запрещенную законодательством РФ.

7. Предложенная методика проверки текста на однородность и поиска заимствований позволяет выявлять неоднородные фрагменты с точностью до 94%, определять авторство научных текстов или их фрагментов с точностью более 90%

* Запрещенная в Российской Федерации экстремистская организация.

и подтверждать авторство с точностью до 99%. Методика учитывает случаи генерации текста моделями 3-го и 4-го поколений.

8. Методология, модели, алгоритмическое и программное обеспечение могут быть использованы при решении смежных задач кибербезопасности и анализа текстовой информации, не рассмотренных в диссертационном исследовании.

Реализация результатов работы

Результаты диссертационного исследования использованы при выполнении следующих проектов:

1. «Программное обеспечение для исследования характеристик текста в задачах идентификации автора» программы ФСРМПНТ «У.М.Н.И.К.» (договор № КР 04/07 от 9.06.2007 г.; № 014/08 от 9.09.2009 г), 2007–2009 гг. – Руководитель.

2. «Методология анализа и обеспечения кибербезопасности объектов критической информационной инфраструктуры», проект № FEWM-2020-0037 в рамках базовой части государственного задания ТУСУРа на 2020–2022 гг. – Исполнитель.

3. «Методология использования методов машинного обучения для обеспечения кибербезопасности», проект № FEWM-2023-0015 в рамках базовой части государственного задания ТУСУРа на 2023–2025 гг. – Исполнитель.

4. «Технология семантической контент-фильтрации сетевого трафика (нежелательного контента)». Грант на государственную поддержку центров Национальной технологической инициативы на базе образовательных организаций высшего образования и научных организаций. Соглашение о предоставлении субсидии от 14.12.2021 г. № 70-2021-00246, 2022 г. – Руководитель.

5. «Прикладные методики семантического анализа текста в сети Интернет», хоз. договор с ООО «СИБ» (Договор № 10-ТДВ от 28.03.2023), 2023 г. – Исполнитель.

6. «Интерфейсы взаимодействия в киберфизических системах», подпроект №18 в рамках стратегического проекта № 2 «ИТ, безопасная цифровая среда и киберфизические системы» госпрограммы «Приоритет 2030», 2022–2023 гг. – Исполнитель.

7. «Кибербезопасность объектов критической информационной инфраструктуры. Безопасный искусственный интеллект», подпроект № 53 в рамках стратегического проекта № 2 «ИТ, безопасная цифровая среда и киберфизические системы» госпрограммы «Приоритет 2030», 2024 гг. – Исполнитель.

Положения, выносимые на защиту

1. Новая комплексная методология идентификации автора текста, учитывающая особенности естественно- и искусственно-языковых текстов, атаки на методы идентификации автора, основанная на классических и современных методах глубокого и семантического анализа, позволяет создавать прикладные инструменты и решать широкий класс задач кибербезопасности, связанных с обработкой текстовой информации.

Соответствует п. 1 паспорта специальности 2.3.6: Теория и методология обеспечения информационной безопасности и защиты информации.

2. Модель создания текста автором в киберсреде позволяет учитывать влияние атрибутов авторов, вида деятельности по созданию текста и специфику среды на семантику текста и авторский стиль, а также особенности решаемой задачи кибербезопасности.

Соответствует п. 1 паспорта специальности 2.3.6: Теория и методология обеспечения информационной безопасности и защиты информации.

3. Методика идентификации автора естественного текста, отличающаяся использованием комбинации GRU+CNN и SVM, обученных на информативных признаках, полученных генетическим алгоритмом и методом на основе регуляризации, позволяет определять автора в случае открытого и закрытого множества кандидатов с точностью до 99% (на 2–14% выше, чем аналоги), устойчива к внедрению имитирующих авторский стиль образцов, созданных генеративными моделями, применима для коротких (до 250 символов) и длинных текстов (от 15000 символов).

Соответствует п. 13 паспорта специальности 2.3.6: Методы и модели выявления и противодействия распространению ложной и вредоносной информации

и п. 12 Технологии идентификации и аутентификации пользователей и субъектов информационных процессов. Системы разграничения доступа.

4. Методика идентификации автора исходного кода программы, отличающаяся использованием нового классификатора на основе CodeBERT, позволяет определять автора исходного кода, написанного на интерпретируемом или компилируемом языке, с точностью до 93% в простых случаях и с точностью до 90% (на 20–30% выше, чем у аналогов) в случаях, осложненных обфускацией, стандартами кодирования, искусственной генерацией или командной разработкой.

Соответствует п. 13 паспорта специальности 2.3.6: Методы и модели выявления и противодействия распространению ложной и вредоносной информации.

5. Методика бинарной и мультиклассовой классификации текстов по возрастным группам автора, отличающаяся использованием метода компьютерного зрения VGG-Face для фильтрации недостоверных данных и fastText для классификации, позволяет повысить точность определения возрастной группы, включая группу до 18 лет, до 82% (на 2% выше аналогов).

Соответствует п. 13 паспорта специальности 2.3.6: Методы и модели выявления и противодействия распространению ложной и вредоносной информации.

6. Методика определения текстов деструктивной и экстремистской направленности, отличающаяся использованием моделей GRU+CNN и BERT, методов семантической кластеризации текста и трансферного обучения, позволяет снизить требуемый объем текстового образца до 250 символов и повысить точность выявления текстов деструктивной и экстремистской направленности до 94% и их распространителей до 95% (на 1–22% выше в сравнении с аналогами).

Соответствует п. 13 паспорта специальности 2.3.6: Методы и модели выявления и противодействия распространению ложной и вредоносной информации.

7. Методика определения пола и гендера автора текста на основе ансамбля SVM, BERT и CNN, отличающаяся определением гендеров ЛГБТ-сообщества*, признанного экстремистским на территории Российской Федерации, позволяет

* Запрещенная в Российской Федерации экстремистская организация.

определять пол с точностью 88–92% (на 9% выше аналогов) и признаки ЛГБТ* в тексте с точностью 93%.

Соответствует п. 13 паспорта специальности 2.3.6: Методы и модели выявления и противодействия распространению ложной и вредоносной информации.

8. Методика проверки однородности текста и поиска заимствований, отличающаяся использованием сиамских сетей SimNN, обученных с контрастивной и тройной функциями потерь, позволяет выявлять неоднородные фрагменты научного текста с точностью 94%, определять их авторство на открытом множестве кандидатов, в том числе генеративные модели, с точностью более 90% и подтверждать автора двух и более текстов с точностью до 99%.

Соответствует п. 13 паспорта специальности 2.3.6: Методы и модели выявления и противодействия распространению ложной и вредоносной информации.

9. Разработанное алгоритмическое и программное обеспечение для анализа естественно- и искусственно-языковых текстов, созданные репрезентативные корпуса текстов позволяют идентифицировать автора и его атрибуты (пол, гендер, возраст, настроение, убеждения), выявлять тексты, нарушающие законодательство РФ, и решать прикладные задачи кибербезопасности.

Соответствует п. 5 паспорта специальности 2.3.6: Методы, модели и средства (комплексы средств) противодействия угрозам нарушения информационной безопасности в открытых компьютерных сетях, включая Интернет.

Внедрение результатов работы

Элементы методологии, разработанное программное обеспечение внедрены в АО «Национальный Инновационный Центр», ООО «СИБ», ООО «НТР», ООО «Сибэдж», войсковую часть 51952, ИШИТР ТПУ, ОБК УМВД России по Томской области, на экономическом факультете МГУ имени М.В. Ломоносова, для решения задач кибербезопасности и обработки текста, включая методики определения автора естественного текста, его возраста, пола и гендера, а также обнаружения текстов деструктивной и экстремистской тематики, определения авторства исходных кодов программ. Внедрение показало положительный эффект, состоящий в сокра-

щении временных затрат, повышении точности и эффективности работы отдельных систем и процессов, минимизации потенциальных финансовых и репутационных потерь за счет применения разработанных в диссертации подходов.

Результаты диссертационной работы используются в учебном процессе Томского государственного университета систем управления и радиоэлектроники при изучении дисциплин «Языки программирования» и «Основы построения систем искусственного интеллекта и машинного обучения». Методика проверки однородности текста и поиска заимствований используется для оценки работ студентов.

Личный вклад автора

Все результаты и положения, составляющие научную основу диссертационной работы и выносимые на защиту, получены автором лично. Соавторы, принимавшие участие в отдельных направлениях исследований, указаны в списке основных публикаций по теме диссертации.

Автором самостоятельно разработаны методология идентификации автора текстовой информации, модель создания текста автором в киберсреде, методики идентификации автора естественно-языкового текста, его возраста, пола и гендера, автора исходных кодов программ, деструктивной и экстремистской направленности в текстах и определения автора таких текстов, проверки однородности текста и поиска заимствований.

Эксперименты проведены с помощью программного комплекса, разработанного по инициативе, под руководством и при непосредственном участии автора, на основе разработанной им методологии, методик, моделей и алгоритмов совместно с аспирантами и студентами ТУСУР научной группы автора.

Апробация работы

Материалы работы докладывались и обсуждались:

– на Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых «Научная сессия ТУСУР», 2006–2024 гг., Томск;

– Международной научно-методической конференции, посвященной 90-летию высшего математического образования на Урале «Актуальные проблемы математики, механики, информатики», 2006 г., Пермь;

– Международной конференции «Interactive Systems and Technologies: The Problems of Human-Computer Interaction», 2007 г., Ульяновск;

– Международной научно-практической конференции «Электронные средства и системы управления», 2007 г., 2009 г., 2013 г., 2015 г., 2017 г., 2018 г., 2019 г., 2020 г. Томск;

– Всероссийской научной конференции «Техническая кибернетика, радиоэлектроника и системы управления», 2008 г., Таганрог;

– Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых с международным участием «Молодежь и современные информационные технологии», 2008 г., 2009 г., Томск;

– Всероссийской научно-практической конференции «Проблемы информационной безопасности государства, общества и личности, безопасность нанотехнологий», 2009 г., Томск;

– Международной конференции по компьютерной лингвистике «Диалог», 2009 г., 2010 г., 2010 г., 2021 г., Москва;

– Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых «Технологии Microsoft в теории и практике программирования», 2010 г., Томск;

– Международной конференции «The II International conference R. Piotrowski's Readings in Language Engineering and Applied Linguistics», 2017 г., 2019 г., Санкт-Петербург;

– XVI Международной научно-практической конференции «Татищевские чтения: актуальные проблемы науки и практики», 2019 г., Тольятти;

– Международной научно-методической конференции «Современное образование: качество образования и актуальные проблемы современной высшей школы», 2019 г., Томск;

– XVI Международной конференции студентов, аспирантов и молодых ученых Перспективы развития фундаментальных наук, 2019 г., Томск;

– Международной конференции «2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)», 2019 г., Томск;

– Международной конференции «YRID-2020: International Workshop on Data Mining and Knowledge engineering», 2020 г., Ставрополь;

– XVII Международной научно-практической конференции «Перспективы развития фундаментальных наук». Город Томск, 2020 г.;

– на XVIII Международной школе-конференции студентов, аспирантов и молодых ученых Инноватика-2022. Город Томск, 2022 г.

Публикации по теме диссертационной работы

Результаты диссертационной работы отражены в 94 публикациях, в том числе в 31 публикации в рецензируемых научных изданиях, рекомендуемых для опубликования основных научных результатов диссертаций на соискание ученых степеней кандидата и доктора наук, из них 18 публикаций в рецензируемых журналах из перечня ВАК и 13 публикаций в изданиях, рецензируемых Web of Science и/или Scopus (из них 7 – в журналах, входящих в первый и второй квартили), а также в 1 монографии, 8 свидетельствах о регистрации программы для ЭВМ и 3 свидетельствах о регистрации баз данных.

Структура и объем работы

Диссертация состоит из введения, шести глав, заключения, списка литературы и 4 приложений. Объем диссертации с приложениями составляет 400 страниц, в том числе 90 рисунков и 59 таблиц. Список литературы включает 334 наименования.

1 ОБЗОР ПРОБЛЕМ И РЕШЕНИЙ В ОБЛАСТИ АНАЛИЗА ТЕКСТОВ ДЛЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ

1.1 Идентификация автора текста и защита интеллектуальной собственности

Одной из проблем в области кибербезопасности является нарушение авторских прав, в том числе личных неимущественных исключительных (интеллектуальной собственности) на результаты труда, представленные в текстовой форме. Это может проявляться в виде незаконного присвоения чужих текстов с целью получения материальной выгоды или в попытках приписать себе авторство чужого творчества. Защита прав на интеллектуальную собственность в цифровой среде зависит от способности противостоять таким нарушениям и предотвратить возможные угрозы. Методы определения авторства позволяют выявить подобные нарушения и установить личность создателя текста.

Автор является ключевой фигурой для установления прав на произведение. Поэтому важно выделить особенности стиля письма всех потенциальных авторов для того, чтобы в дальнейшем, во-первых, установить, присутствует ли автор в списке кандидатов, а во-вторых, в случае присутствия, с высокой точностью выявить его из множества кандидатов.

Согласно современному институту авторского права, результаты интеллектуального труда являются объектами авторской собственности, следовательно, авторские права на них должны быть защищены соответствующими нормативными актами. Ключевые изменения в данной области законодательства были вызваны масштабной цифровизацией и повсеместным использованием, созданием, хранением и передачей текста в цифровом формате.

Неизменным остается факт заинтересованности создателя произведения в широком распространении его творчества с целью извлечения материальной выгоды или получения популярности.

Авторы, самостоятельно публикующие свои текстовые произведения в сети Интернет, могут ограничить контент от широкой аудитории, однако, в случае получения материальной выгоды от просмотра и чтения текстов, в интересах самого автора, наоборот, популяризировать свои произведения. В случае открытого доступа, присутствует возможность как несанкционированного присвоения авторской деятельности, так и попытки имитации стиля письма с целью размещения текстов от имени автора без его согласия. Лингвистические экспертизы, являющиеся наиболее популярным способом решения таких проблем, затруднительны ввиду ограниченности человеческих ресурсов (времени экспертов, числа квалифицированных специалистов, высокой стоимости экспертиз). Высокоточные автоматизированные системы идентификации автора решают данные проблемы и могут привлекаться в роли рекомендательных систем при принятии решений.

Атрибуция текстов является важной проблемой в криминалистике, где знание истинного автора анонимного текста может облегчить и ускорить работу правоохранительных органов. В связи с переводом многих видов коммуникации в режим онлайн, в обращении стремительно возрастает количество печатных текстов, их экспертизы проводятся значительно чаще, чем рукописных. Предметом споров могут являться как авторство предсмертных записок, так и право на объект интеллектуальной собственности. В таких случаях проводят автороведческие экспертизы. Количество экспертов, обладающих необходимыми для проведения экспертизы знаниями, невелико. Из-за недостатка количества экспертов и стоимости проведения возникает проблема получения быстрого результата экспертизы в провинциальных городах. Решением может стать создание инструментария, способного выступать в качестве рекомендательной системы при проведении экспертиз, существенно сокращающего время принятия решение при большом количестве кандидатов.

Таким образом **проблема идентификации автора текста для защиты интеллектуальной собственности является важной задачей кибербезопасности.**

Подробный обзор статей по теме идентификации автора текста до 2010 года представлен в диссертационной работе автора [289] и монографии [299].

Подчеркивается важность работ Н.А. Морозова [273], В.П. Фоменко и Т.Г. Фоменко [328], М.А. Марусенко [264], Д.В. Хмелева [331], О.Г. Шевелева [334], А.А. Рогова [286], А.Ю. Комиссарова [255] и др. Создана методика идентификации автора текста на основе ансамбля метода опорных векторов (SVM), нейросетевого подхода и методов сглаживания (рисунок 1.1). Были получены информативные текстовые признаки, отражающие уникальный стиль автора. Исследование показало, что методика обеспечивает точность определения авторства в диапазоне 95–98% при анализе трех текстовых образцов каждого автора, каждый из которых содержит 20000 символов, при условии 5 или 10 потенциальных авторов. Для коротких текстов до 100 символов с двумя возможными авторами, точность составила в среднем 76%.

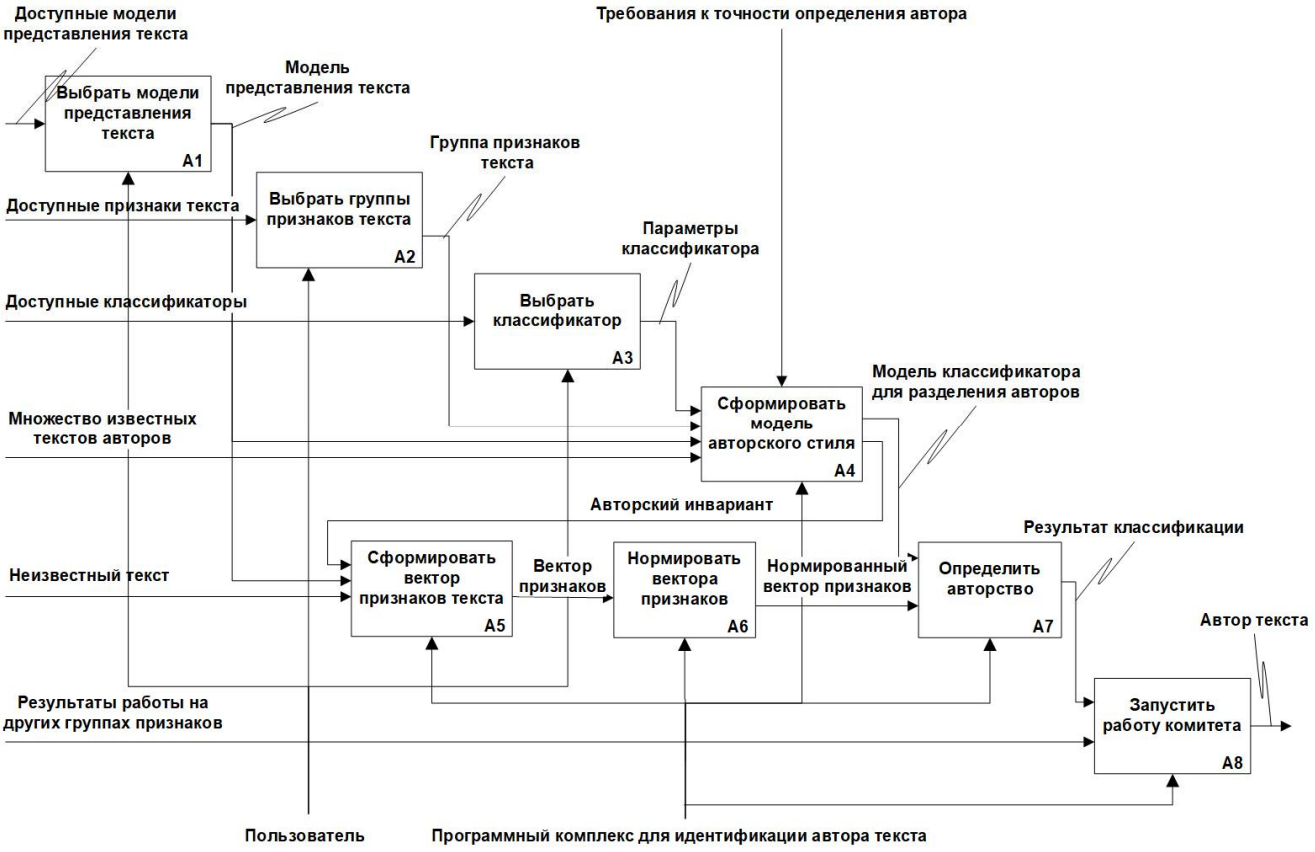


Рисунок 1.1 – Ранняя методика идентификации автора Романова А.С.

Методы идентификации автора можно разделить на методы статистического анализа [70, 90, 94, 99, 109, 167] и машинного обучения [4, 71, 96, 111, 141, 208,

215, 218, 287]. Однако в последнее десятилетие на первый план выходят семантические методы глубокого анализа [32, 33, 38, 71, 111, 152, 176, 208, 218, 222]. Сводная таблица, содержащая более полный анализ основных работ в области определения автора текста, приведена в приложении А (таблица А.1).

1.1.1 Методы статистического анализа

В статье А.С. Сурковой [321] описана методика на основе информационных портретов, сформированных через вычисление взаимной информации между признаками текстов. Исследования проводились на 20 авторах. Основная задача исследования заключалась в разработке метода, позволяющего идентифицировать автора текста на основе статистической зависимости между различными элементами текста. Используемыми методами выбраны деревья решений, а также методы, основанные на моделях сжатия текста и энтропии. Признаковое пространство включало биграммы букв и символов. Эксперименты показали точность 88%.

В работе [246] А.Р. Дубовик исследует возможность определения авторства текстов различных стилей на основе статистических параметров. Автором разработана компьютерная программа анализа русскоязычных текстов, принадлежащих к научному, художественному, официально-деловому и публицистическому стилям, с использованием морфологической аннотации. Каждый корпус включал 500 тысяч словоупотреблений. При анализе учитывались такие признаки как средняя длина предложения, соотношение динамичности и статичности, а также частоты использования различных частей речи. Точность классификации художественных и деловых текстов составила 88%. Научные тексты были классифицированы с точностью 74%, а публицистические – с точностью 56,9%.

В работе В.В. Исаченко [250] описана разработанная система для устранения неоднозначностей в авторстве англоязычных статей с применением русскоязычных баз данных. Эта система направлена на обнаружение и коррекцию ошибок при определении авторства научных работ, что может способствовать более точному

поиску публикаций определённого автора и корректному подсчету его индекса цитирования. В качестве основного ресурса для публикаций использовалась база данных link.springer.com, а для верификации данных об авторах и их работах – научная электронная библиотека eLIBRARY.ru. В эксперимент были включены статьи 25 сотрудников ИСИ СО РАН, присутствующих в базе link.springer.com. Система смогла правильно идентифицировать 79% публикаций авторов в ограниченной тематической выборке. В кросс-тематической выборке точность составила 92%.

М.Ю. Кислицына и Ю.Н. Орлов [253] исследовали различные комбинации символьных триграмм в анализе авторства русскоязычных текстов, а также переводных зарубежных произведений в сочетании с методом *k*-ближайших соседей (KNN). При сборе набора данных использовались только авторы, имеющие 5 и более художественных произведений от 10000 символов каждое. Итоговый набор включал 108518 текстов 8287 авторов, из которых 66215 текстов написаны на русском, остальные являются переводами иностранных произведений. Тексты относятся к различным тематическим разделам (художественные, боевики, детективы, любовный роман, современная проза, детская литература). При проведении экспериментов использовались 10 авторов. Авторы не разграничивали оригинальных писателей и переводчиков. При анализе на основе символьных *n*-грамм в случае униграмм точность составила 26,9%, биграмм – 65,8%, триграмм – 73,2%. Ошибка по жанрам варьируется от 9% (фэнтези) до 26% (любовный роман). Проведена классификация ошибок для разработки метода коррекции, на основе чего установлено, что в зависимости от выбранной комбинации триграмм символов, величина ошибки меняется до 15%.

А.Н. Ульченко в статье [324] исследует методы векторизации текста в задаче определения автора художественного произведения. Подход включает применение инструмента 2B-PLS (Two-Block Projection to Latent Structure) и программного пакета JACOBI 4. Набор данных включал поэзию Цветаевой и Ахматовой, в общей сложности 310 текстов. Среди частотных признаков отмечены распределения глаголов, существительных, прилагательных, наречий, знаков пунктуации

и служебных частей речи, а также средние длины строк и их число, количество знаков препинания. В качестве метаинформации автор использовала год создания текста. Полученные векторы сравнивались метрикой косинусного сходства. Отмечено, что предложенный подход позволяет выявить тексты с наибольшим показателем однородности, идентифицировать автора стихотворного текста с точностью до 81%.

Судебные эксперты подчеркивают важность адаптации критериев для определения авторского стиля письма в контексте атрибуции текстовых материалов в Интернете. В переписке пользователи часто прибегают к невербальным средствам, таким как стикеры и эмотиконы, которые могут изменять или дополнять смысл сообщения. При этом правила орфографии и пунктуации перестают соблюдаться. Криминалисты также указывают на значимость определенных текстовых признаков, таких как эрративы, или умышленные ошибки. Основная идея, предложенная Т.П. Соколовой [319], заключается в смещении фокуса на семантические аспекты текста. Это поможет не только различать авторов по количественным мерам, таким как частотные распределения и статистические данные, но и выявлять главные идеи автора, что улучшит качество проведения судебных экспертиз.

1.1.2 Методы машинного обучения

Работа Е. Пимоновой и др. [171] посвящена проблеме исследования авторского стиля в русскоязычных текстах. В частности, решалась задача установления авторства на базе данных 30 авторов, 1506 текстов, написанных в период XVIII – XXI веков. Были применены различные подходы к решению проблемы атрибуции: случайный лес (RF), логистическая регрессия (LR) и SVM. Авторы исследовали тексты на трех языковых уровнях: лексика, морфология и синтаксис. Проведенные эксперименты показывают эффективность автономного использования признаков, а также повышение качества классификации при совместном использовании признаков наряду с классическим подходом на основе Doc2Vec. Последний подход

показал точность 93%. После настройки параметров Doc2Vec, удалось повысить точность до 98%. Вторыми по эффективности названы простые синтаксические признаки. Точность при использовании таких признаков и LR составила 89%.

В статье П. Паничевой и Т. Литвиновой [164] предложено решение задачи определения автора с учетом тематических направлений коротких текстов. Авторы сравнивали тексты по схеме «каждый с каждым», используя меры, основанные на расстоянии и SVM, а также *n*-граммы, семантическую кореферентность и части речи. Для проведения экспериментов были отобраны 120 текстов пяти авторов, относящиеся к трем тематикам. Исследователи выделили 17 из 300 признаков наиболее значимыми и получили точность от 60 до 88%.

С.В. Чащин [333] представил программный продукт StylometRy, позволяющий определить автора спорного текста. Текст представляется в виде модели мешка слов, особое внимание уделяется местоимениям. В качестве классификаторов были выбраны методы наивного байесовского классификатора (NB), KNN и LR. Проверка моделей осуществлялась на текстах Л.Н. Толстого, М. Горького, А.П. Чехова. Точность модели для текстов объемом свыше 150 000 символов находилась в пределах 60–100% (среднее – 87%). Минимальный объем текста для проведения анализа составил 5500 слов.

Среди семантических методов распространена фреймовая семантика [157, 101]. В лингвистике данным термином называют теорию о понимании значения текста с использованием ряда слов, связанных по смыслу. Для каждого слова формируется семантический фрейм путем создания цепочки на основе энциклопедических значений – согласно словарю, для каждого понятия есть слово (источник), на которое оно ссылается (или «выделяет» – в терминологии фреймовой семантики). Кроме того, слова в структуре фрейма представлены цепочкой – формируется некоторая перспектива принадлежности, в которой рассматривается фрейм. Например, связанные между собой глаголы «написать» и «прочитать» рассматривают со стороны автора и читателя, соответственно. В терминологии семантических фреймов такое явление называется асимметрией. Применительно к анализу

естественного текста, фреймовая семантика распространяется не только на лексемы, но и другие уровни текста.

Ряд исследований направлен на классификацию авторов научных работ.

В работе [254] рассматривается проблема точности расчета индекса цитируемости и плагиата на основе 10 ключевых слов научных публикаций с известным авторством и определения схожести по ним со спорными научными трудами. Наиболее близкие друг к другу векторы означали статьи, написанные одним и тем же автором. Авторы отмечают, что ключевые слова, отмеченные создателями публикаций и выделенные при помощи LSA (латентно семантический анализ), совпадают только в 60% случаев.

О.М. Атаева и соавторы [233] изучали идентификацию автора текста как задачу определения авторского вклада в публикации в электронных библиографических платформах. Отмечаются следующие сложности: авторы-двойники с полным совпадением ФИО, самоцитирования, автоплагиат, ошибки цифровых платформ и непосредственно плагиат. Публикация представляется в виде массива, элементами которого служат структурные части статьи: аннотация, ключевые слова, разделы и список авторов. В исследовании использовались научные публикации, связанные с высшей математикой. Для выделения ключевых слов (слов и словосочетаний с наибольшим семантическим значением) использовался расчет меры TF-IDF [203]. Термины, для которых мера была меньше порогового значения, исключались. Важно отметить, что выделение терминов происходило в контексте. Под контекстом понимается окно из *n*-слов для текущего слова (для которого строится вектор). Для выделения контекста использовалась модель Word2Vec. Разрешение противоречий об авторском праве осуществляется путем оценки множества связей всех элементов вторичной информации о публикации. Результатом сравнения может быть добавление автора в некоторую предметную область, т. е. расширение тезауруса адресата и персонального тезауруса автора, или появление в библиотеке полных тезок, но из разных областей знаний. Показано, что современные средства анализа данных позволяют оценить вклад автора в публикацию. Однако реальный вклад в научное исследование может оценить только научное сообщество.

Недостатками исследования являются отсутствие подробной информации о базе данных (количество авторов и публикаций) и количественных результатов классификации. Также не объяснено, как авторы предлагают решать проблему соавторства и выделения частей, написанных каждым из них.

1.1.3 Методы глубокого обучения

В последнее десятилетие все большую популярность приобретают методы, основанные на глубоких НС [235], вытесняя классические методы машинного обучения, а также методы на основе семантических характеристик [71].

Диссертация Н.Д. Москина [274] включает исследования, направленные на разработку и усовершенствование передовых теоретико-графовых моделей для задачи определения авторства спорных текстов, включая технологию нейронных сетей GNN (Graph Neural Network), которые интегрируются в систему СМАЛТ, описанную в монографии автора и его коллег [287]. В работах анализировались фольклорные и литературные произведения. В исследовании автор оперирует агрегационными подходами при использовании графовых моделей, учитывая их иерархичность, неопределенность и временные аспекты. Н.Д. Москин предложил формулу расчета расстояния на основе максимального общего подграфа, а также применил непараметрический ранговый критерий Краскела – Уоллиса для проверки гипотез о связи жанра и структуры графа. Разработанный ансамбль теоретико-графовых моделей позволяет детально описывать тексты, рассматривая их сложную языковую структуру и динамику. Программное обеспечение «Фольклор» (рисунок 1.2) адаптировано под особенности текстов XIX века, включая создание специализированного словаря старорусского языка. В ходе экспериментов использовались такие методы, как рекуррентные НС (RNN), сети долгой краткосрочной памяти (LSTM), Transformer, а также SVM и RF. Для оценки использовались фрагменты текстов объемом не менее 10000 слов. Векторизация осуществлялась с использованием

Word2Vec. Лучшие результаты показала модель Transformer с параметрами: размерность эмбеддинга – 32, количество скрытых слоев – 16, слои самовнимания – 2, размерность батча – 32. Точность модели составила 97%, тогда как деревом решений был достигнут минимальный результат – 43%.

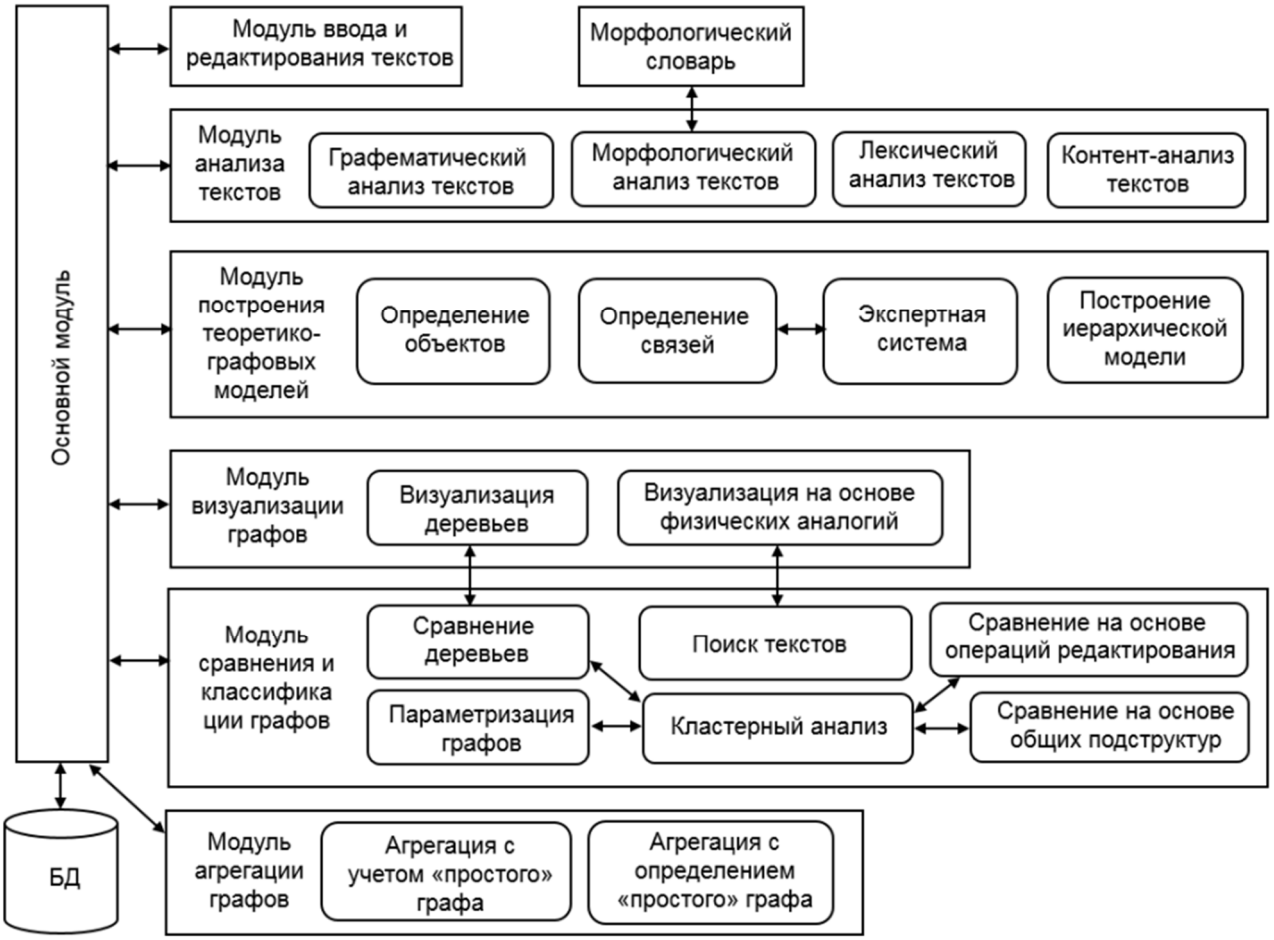


Рисунок 1.2 – Структура информационной системы «Фольклор» Н.Д. Москина

В статье [152] представлен подход к проверке авторства с помощью архитектуры Transformer, сочетающей стилистический и семантический анализ. Предлагаемый подход использует модель трансформации текста, которая принимает на вход два текста и предсказывает, написаны они одним и тем же автором или нет. Подход сочетает в себе два вида анализа: стилистический анализ, который фокусируется на стиле письма автора, и семантический анализ, который фокусируется на семантическом содержании текстов. В статье также представлен всесторонний анализ результатов, включая сравнение производительности модели на различных типах текстов, таких как электронные письма и романы, и анализ вклада различных

компонентов модели в ее общую производительность. На корпусе PAN 2018 предложенный подход показал *F1*-score 0,86, что выше, чем *F1*-score, полученный базовым методом с наилучшими показателями. На корпусе отзывов Amazon предложенный подход достиг точности 0,98.

В работе [38] определение автора текста осуществлялось с помощью сверточной НС (CNN). Корпус включал 20 авторов, написавших 400 статей по машинному обучению. Модель, предложенная авторами, на верхнем уровне представляет собой многослойную CNN, которая либо вычисляет вероятностное распределение для всего текста, либо среднее распределение вероятностей по отдельным предложениям текста (несколько кандидатов-авторов). На нижнем уровне тексты приводятся к единой длине 128 символов. Далее используются Word2Vec и Global Vectors for Word Representation (GloVe). Модель авторов сравнивалась с SVM и многослойным перцептроном (MLP). Для этих моделей применялись *n*-граммы. Результаты показали, что CNN (на основе Word2Vec+GloVe), разработанная авторами, достигла точности 65%, что является лучшим результатом (SVM – 45,3%, MLP+Doc2Vec – 40%).

A. Abbasi и др. [2] представили сравнение ансамбля XGBoost, RF и многослойного перцептрона с моделью Multi-depth DistilBERT. В исследовании использовался набор данных из 143000 новостных статей 16408 авторов. Для извлечения признаков применялись методы Count Vectorizer и TF-IDF. Для обучения моделей использовалось 350 образцов каждого автора, для тестирования – 150. С помощью ансамблевого решения удалось достичь точности 97% и 79% для 10 и 20 авторов, соответственно, что превосходит результат Multi-depth DistilBERT на 9%.

В статье [106] целью исследования было сравнение больших языковых моделей семейства GPT (GPT-4 Turbo, GPT-3.5 Turbo) с моделями семейства BERT (BERT, ELECTRA, DeBERTa) для определения авторства текста. Авторы статьи предложили Linguistically Informed Prompting (LIP) – метод формирования промптов, который предоставляет текст с акцентом на конкретные лингвистические признаки. В отличие от традиционных подходов, которые требуют сложной настройки и выделения признаков, LIP позволяет модели эффективно работать в условиях,

когда она не была специально обучена для определенной цели (zero-shot) и справляться с задачами на данных, которые относятся к различным доменам (cross-domain), повышая как точность, так и объяснимость решений. Ключевые признаки, выделяемые промптом, включали фразовые и модальные глаголы, пунктуацию, редкие слова, суффиксы, цифры, наличие признаков сарказма, орфографические ошибки и опечатки. Для 10 классов GPT-4 Turbo достигла точности 88,9% для набора данных электронной переписки и 86,7% для набора данных постов. При классификации с разбиением на 20 классов точность снизилась: 77,2% и 60,5% для тех же наборов, соответственно. Точность BERT-моделей оказалась ниже, чем результаты GPT-4 Turbo на 20–30%.

В статье [158] авторство текстов определялось для румынского языка, как языка, для которого существует мало обучающих данных, с помощью гибридного подхода на основе архитектуры Transformer. Методика объединяет информативные лингвистические признаки с эмбеддингами предварительно обученной модели RoBERT, учитывающими контекст. Используемый набор данных состоит из текстов 19 авторов, включая 1263 рассказов. В качестве лингвистических признаков используются частоты слов, длины предложений и синтаксические структуры, с последующим отбором признаков с использованием среднего ранга Краскела – Уоллиса. Предложенная модель достигает F-меры 0,87 для набора с 19 авторами.

Основные выводы из анализа методов идентификации автора следующие. Среди методов машинного обучения продолжает превалировать метод SVM. Однозначных выводов о лучшей архитектуре глубоких моделей применительно к решаемой задаче нет. Методы, на основе архитектур Transformer и BERT [18, 30], являющиеся state-of-the-art в области текстового анализа, в русском сегменте почти не рассмотрены.

1.2 Определение пола и гендера, возраста автора. Выявление признаков ЛГБТ* и педофилии

К числу характеристик, влияющих на авторский стиль, относят пол, возраст, уровень образования, профессию и квалификацию, настроение, точку зрения по определенному вопросу и др. Первичными атрибутами являются пол и возраст, поскольку именно они в большей степени влияют на личность человека.

Совокупность атрибутов формирует уникальную языковую личность. Одним из направлений практического применения методик определения пола и возраста является криминалистика, где важно определение психологического портрета преступника и профилирование автора. Это понятие на сегодняшний день широко используется при теоретическом обосновании заключений криминалистических экспертиз [234], классификационных, психологических и маркетинговых исследованиях, социолингвистических идентификациях [238, 248, 284,], диагностировании личности по тексту [302], определении психологического профиля поступающих на службу в ряды МЧС России [263], а также в других прикладных областях.

В контексте проблем кибербезопасности подобные методики являются основой для мониторинга социальных сетей для выявления информации, пропагандирующей нетрадиционные сексуальные отношения и предпочтения, педофилию, смену пола. Важной проблемой также является недопущение детей и подростков до запрещенного или шокирующего контента, который имеет возрастное ограничение «18+», либо ограничение общения с целью предотвращения педофилии.

Для защиты населения в киберпространстве вносятся соответствующие поправки в законодательство Российской Федерации.

Законом № 478-ФЗ от 05.12.2022 в ряд законодательных актов внесены изменения, запрещающие распространение любой информации, которая «пропагандирует порнографию, насилие и жестокость, содержит нецензурную брань, пропагандирует нетрадиционные сексуальные отношения и (или) предпочтения, педофилию, смену пола», среди людей всех возрастов. Корреспондирующие изменения

* Запрещенная в Российской Федерации экстремистская организация.

в соответствии с федеральным законом № 195-ФЗ внесены также в Кодекс Российской Федерации об административных правонарушениях: статьей 6.21 КоАП РФ предусмотрена ответственность за указанные действия в следующем размере: штраф от 50 000 до 400 000 руб. для физ. лиц, от 100 000 до 800 000 руб. для должностных лиц, до 5 млн руб. юридическим лицам».

Решением Верховного Суда Российской Федерации от 30.11.2023 по делу № АКПИ23-990С «движение ЛГБТ» признано экстремистской организацией и запрещена его деятельность на территории России.

Таким образом **определение пола и возраста по тексту для дальнейшего выявления признаков пропаганды ЛГБТ и педофилии является важной задачей кибербезопасности**, актуальным является также вопрос создания программного обеспечения для мониторинга социальных сетей.

Сводная таблица, содержащая анализ основных работ в области определения пола и возраста автора текста, приведена в приложении А (таблица А.2).

1.2.1 Определение пола и возраста автора текста

К настоящему времени существует немало исследований, как для русского, английского, так и других языков [19, 23, 100, 147, 173, 182]. Методы и признаки текста должны учитывать специфику и объем текста, его жанр, тему, источник публикации и интересующие исследователя атрибуты автора в контексте решения прикладных задач [28, 50, 58, 76, 82, 156, 197, 209].

Наиболее полно вопрос определения пола и возраста автора русскоязычного текста освещается в диссертационной работе А.Г. Сбоева [305] и в статье [155]. Сбоев представил методику, основанную на сочетании глубокого обучения, частот последовательностей символов от триграмм до октаграмм, частеречном и тональном анализе, классических векторных представлений и эмбеддингов, а также сиамской НС. Для классификации по полу и возрасту используются SVM и RF, а также глубокие НС на основе рекуррентных и сверточных слоев, в том числе

авторская архитектура SyntGraphLSTM (рисунок 1.3). Особенностью исследования является оценка влияния на результат намеренных искажений авторами своего возраста и пола. Метрика *F1* при определении пола составила 86%, но в случае намеренного искажения опустилась до 64%. Точность определения возрастной группы составляет 48%. Если возраст автора пытаются изменить, то определить в каком направлении это делается (младше или старше) удается с точностью 80%, а сам факт попытки искажения с точностью 44%.

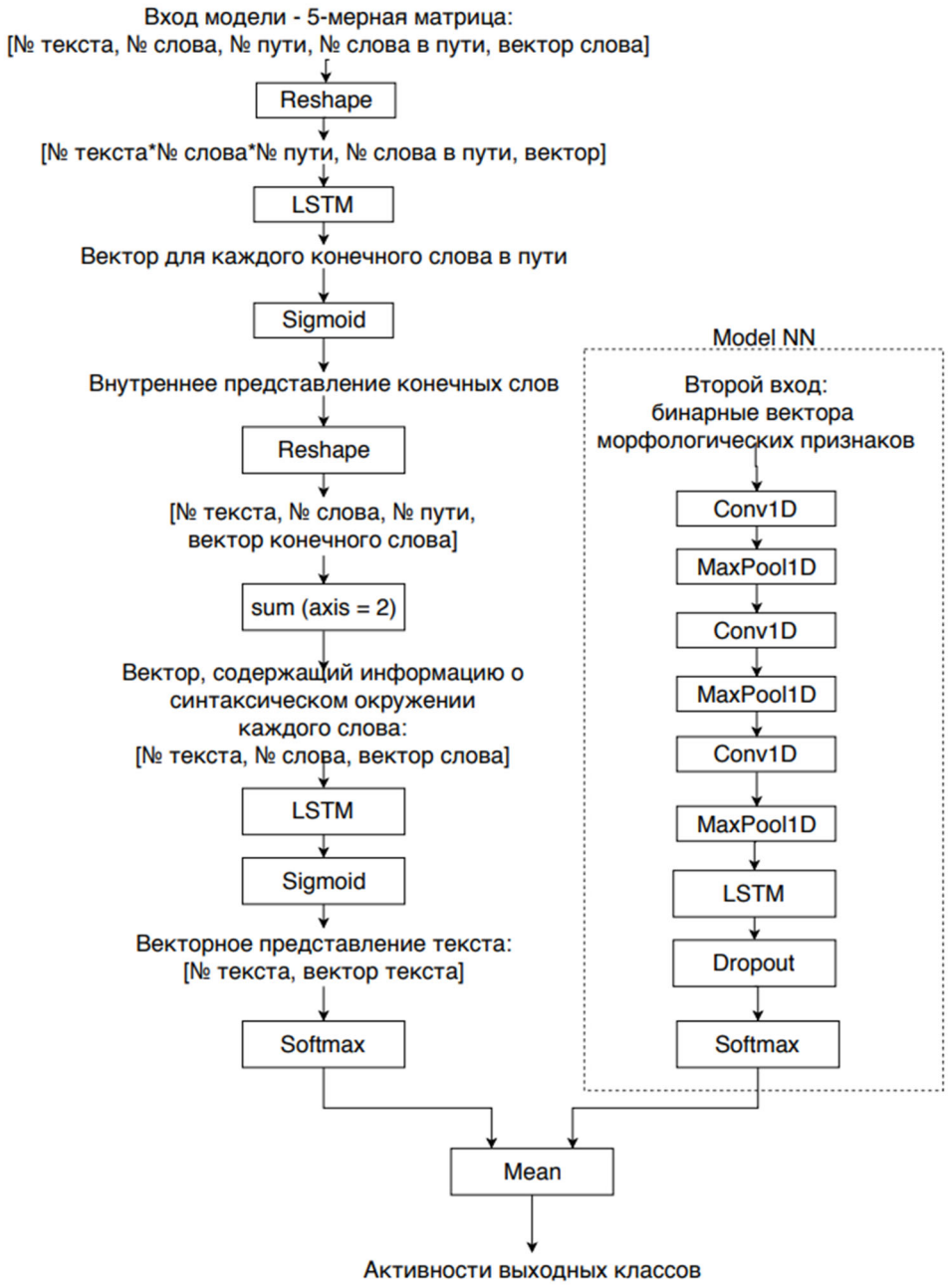


Рисунок 1.3 – Архитектура комбинированной НС SyntGraphLSTM А.Г. Сбоева

Подход к определению пола автора русскоязычного текста представлен Д.А. Галкиной [242]. Отмечается, что пол автора может подсказать окончание глагола в форме прошедшего времени единственного числа. Однако такой подход неприменим для текстов, где повествование идет не от первого лица, и официальных текстов с использованием безличных форм. Поэтому автор опирался на работу [85], где отмечено, что обилие местоимений в тексте может быть характерной чертой женских текстов. Среди применяемых классификаторов для нехудожественных текстов лучше всего себя показал RF – 94%, для художественных – AdaBoost с точностью 85%.

И.В. Огорелков в диссертационной работе [277] рассматривал вопросы исследования гендерных различий мужской и женской письменной речи. Исследование направлено на классификацию тематических текстов, посвященных политике (500 написаны мужчинами и 500 – женщинами, не менее 8000 слов). На первом этапе выделяются смысловые, текстологические, языковые, психолингвистические признаки. Затем добавляются особенности, позволяющие устанавливать характеристики автора. Среди информативных признаков [278] выделены союзы, вводные слова и местоимения. Последние больше распространены у женщин, чем мужчин. Автор не уточняет конкретные данные о корпусе и методах определения пола, однако приводит следующие выводы. К информативным признакам мужчин относят лаконичность и структурность, конкретность, предметность и аргументированность изложения, использование обсценной лексики. Для текстов, написанных женщинами, характерны многословность, гиперболизированность и ассоциативность, использование слов с уменьшительно-ласкательными суффиксами, хаотичность и эмоциональность.

В статье А.Б. Хазовой [330] указывается, что информативными для женских текстов являются частоты знаков пунктуации, особенно восклицательных и вопросительных знаков, а для мужских – частоты существительных, глаголов и предлогов. В результате удалось получить 75% точность идентификации мужчин и 68% женщин с помощью SVM.

Отдельный эксперимент в работе А.К. Ковалёва [268] был посвящен влиянию возраста автора на степень агрессивности. Авторы разделялись на группы 10–20, 21–30, 31–40 лет, а также по гендеру. При определении пола были использованы такие же классификаторы, что и при определении авторства. При этом удалось достигнуть точности 68% при определении пола и 55% при определении возраста автора.

Определение пола автора текста является актуальной темой соревнований PAN. В рамках соревнований 2017 команде под руководством И. Маркова [144] удалось достичь максимального результата для наборов Twitter*, Facebook** и отзывов. Участники разработали подходы, основанные на сочетании SVM и методов математической статистики. Предобработка текстов в обоих случаях включала удаление упоминаний других пользователей, приведение к нижнему регистру, форматирование пробельных и специальных символов. При использовании SVM рассчитывались частотные распределения униграмм слов, *n*-грамм символов, суффиксов и окончаний. При статистическом анализе составлялся словарь уникальных слов на основе обучающей выборки каждого класса, в тестовом образце считалось число совпадений, на основе чего принималось решение о поле автора анонимного текста. Подход на основе SVM позволил добиться точности 93% на наборе данных Facebook** и 62% в случае отзывов. Статистический подход оказался результативным для набора Twitter* (точность 68%). Средняя точность составила 65%. Отметим, что все результаты получены на небольших датасетах (94-776 документов). Подход R. Bhargava [31] и др. был объявлен лучшим при классификации тематических эссе. Команде удалось достичь точности 78%, применяя комбинацию LSTM и BiLSTM, предварительная обработка данных была аналогична команде Маркова и дополнительно включала стемминг и лемматизацию.

В рамках соревнований PAN 2019 года участникам был предложен датасет постов пользователей Twitter* (4800 пользователей в задаче бот-человек и 6720 в

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.
** Деятельность организации Meta Platforms Inc, ее продукта Facebook запрещена в Российской Федерации.
* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

задаче определения возраста, на каждого пользователя приходилось ровно 100 текстов на английском или испанском языках) [82]. Для усложнения задачи были добавлены тексты, созданные ботами. Подобные боты зачастую используются, чтобы привлечь внимание к определенным событиям, в том числе для распространения ложной информации. Задача состояла в определении, создан ли текст человеком или генеративным алгоритмом и в определении пола. Команда победителей под руководством J. Pizarro [173] использовала SVM, TF-IDF и *n*-граммы символов. Точность классификации составила 83% и 81% для текстов на английском и испанском языке.

Определение пола автора для набора PAN-2019 рассмотрено в работе [212]. Авторский статистический подход основан на частотных признаках. В качестве классификатора применялся RF. Всего для обучения модели было использовано 17 признаков на основе стилометрии (частоты специальных символов, знаков пунктуации, коротких слов до 5 символов, длинных слов от 11 символов) и метаданных (число действий с твитом – количество ретвитов, добавлений в избранное, комментариев). Для набора данных на английском языке точность составила 92,45% для задачи обнаружения ботов и 90,36% для гендерной классификации; 89,68% и 88,88% для испанского набора данных соответственно.

Авторы статьи [10] решали задачу определения пола на основе коротких текстов, созданных мужчинами и женщинами, а также генеративными моделями, имитирующими мужской и женский стиль письма. Набор данных состоял из 6000 твитов, включая созданные ИИ. Методика включает предварительную обработку твитов, извлечение TF-IDF и применение отбора признаков с использованием теста Хи-квадрат. В качестве классификаторов использовались SVM, DT, RF и MLP. Лучших результатов удалось добиться с помощью MLP – точность составила 90%.

Статья [226] направлена на определение пола авторов художественных текстов. Используемый набор данных BookSCE содержит 8222 книги. Методика объединяет классические методы машинного обучения (LR, XGBoost и SVM) и модели глубокого обучения (BERT, GPT2, XLNet, RoBERTa и T5). Модели обучаются на предварительно обработанных текстовых данных с использованием как *n*-грамм

слов в качестве признаков (для классических методов), так и контекстных вложений (для моделей глубокого обучения). Результаты исследования показывают, что GPT2 превосходит все другие модели с точностью 92,5%, за ней следует XLNet с точностью 90,7%.

В статье [129] приводится сравнение методов нулевого обучения и комбинации fastText+RoBERTa в задаче определения возраста автора текста. Набор данных включает 14343 текста на английском. Для методов нулевого обучения использовались лексические и синтаксические признаки, такие как частота употребления определенных грамматических конструкций и стиль написания, при использовании RoBERTa признаки извлекались fastText. Классификация осуществлялась на три возрастные категории: до 25 лет, от 26 до 40 лет и старше 40 лет. Результат - точность 88% при совместном использовании fastText и RoBERTa, точность методов нулевого обучения не превышала 77%.

1.2.2 Определение признаков ЛГБТ* в тексте

Тема определения ЛГБТ* на основе текстовой информации находится в самом начале своего развития ввиду недавнего появления самого термина. Касательно анализа русскоязычного текста, на начало 2025 года существует всего несколько работ, связанных с темой.

В исследовании [166] использовались страницы активистов ЛГБТ-сообщества* и политических деятелей на русском, английском и испанском. Всего более 30 000 образцов, по 10 000 текстов для каждого языка. Задача состояла в определении ЛГБТ*-деятелей. В качестве модели применялась LR, точность для всех языков составила более 70%. Максимальный результат был достигнут для английского – 83%.

Авторы исследования [24] определяют категорию запрещенного или негативного контента, например, ЛГБТ*, онлайн-преступления, насилие, вооруженные

* Запрещенная в Российской Федерации экстремистская организация.

действия, распространение порнографии, токсичный текст и т.д. Набор данных был создан с помощью генеративных моделей и включал 124 597 образцов на русском языке. Все тексты были вручную размечены на 21 категорию. Для классификации использовалась модель BERT в имплементации для русского языка. Результаты определения запрещенного контента – от 40% до 85% в зависимости от категории.

В мировом сообществе тема определения ЛГБТ** достигла большего развития. Ученые изучают данное явление как отдельный феномен, так и в совокупности с общим социолингвистическим портретом [201, 217].

Исследование [119] посвящено определению пользователей, относящихся к ЛГБТ** в сравнении с гетеросексуальными респондентами на примере коротких текстов на английском из Twitter*. Для экспериментов были отобраны 16 000 аккаунтов. В качестве признаков использовали биографические характеристики аккаунтов. Подход авторов на основе наивного байесовского классификатора дает многообещающий результат с точностью 88%.

Работа [143] направлена на обнаружение оскорбительного контента в материалах англоязычных ЛГБТ-сообществ**. Набор данных содержит 990 комментариев. Предварительная обработка включала конвертацию текстов в векторную форму на основе BERTVectorizer. Авторы сравнивали модели BERT и ее модификацию, XLM-RoBERTa. Последняя достигла точности 93%, а классическая модель BERT уступила на 2%.

Результаты подчеркивают ограничения современных подходов установления пола и возраста автора на основе текстов. Отношение к полу и возрасту как к фиксированным переменным игнорирует разнообразие способов, с помощью которых человек проявляет свою уникальность средствами естественного языка. Приближение возраста и пола к социальным переменным позволит проводить более тщательный анализ и создавать надежные системы.

** Запрещенная в Российской Федерации экстремистская организация.
* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

Общим недостатком всех рассмотренных подходов является строгая зависимость от длины анализируемого образца и источника данных. Методики предназначены для работы только с одним из типов текстов. В результате модели имеют низкую обобщающую способность и невысокую точность. Также к недостаткам можно отнести неспособность некоторых подходов корректно обрабатывать неформальные тексты пользователей социальных сетей и мессенджеров. Комментарии и сообщения отличаются преднамеренным и случайным несоблюдением норм языка, использованием сленговой лексики и слабой когерентностью. Большинство подходов ориентированы на бинарную классификацию (дети и взрослые), и не предназначены для определения точной возрастной категории, что может иметь значение для автороведческих экспертиз. Также следует учитывать другие атрибуты автора, включая пол, образование и др., поскольку сверстники с разным жизненным опытом и уровнем знаний отличаются связностью изложения мыслей. Стоит принимать во внимание возможность преднамеренного искажения информации автора о себе, а также его психоэмоциональные характеристики.

1.2.3 Модели компьютерного зрения для определения возраста автора

Важным этапом машинного обучения является стадия предварительной обработки данных. Излишне зашумленные или идеально выверенные данные могут негативно сказаться на обучении модели и её предсказательной способности.

Но даже качественная подготовка данных не обеспечивает высокую точность модели. Например, распространены случаи, когда пользователи социальных сетей специально искажают реальный возраст при заполнении анкеты регистрации. Такие данные, собранные для обучения модели, определяющей возраст, становятся непригодными. Отсюда возникает необходимость фильтрации данных.

Решением этой проблемы может **стать применение моделей компьютерного зрения, предназначенных для смежной задачи определения возраста**

пользователя по фотографии, большая часть из которых основана на сверточных архитектурах НС и их модификациях.

Авторами статьи [84] была разработана модель GRA_Net (Gated Residual Attention Network) для определения пола и возраста человека на основе фотографии. GRA_Net комбинирует сети остаточного внимания, маскирование, пространственные и канальные механизмы внимания. Модель обучена и протестирована на 5 наборах данных: FG-Net (1002 фото, 0–69 лет), AFAD (160000 фото, 15–40 лет), набор данных из Википедии (62328 фото, 5–80 лет), UTKFace (20000, 0–116 лет), AdienceDB (26580, 0–60 лет). Результаты определения пола оценивались по метрике точности, возраст определялся с точностью до года, для оценки ошибки модели использовалось среднее абсолютное отклонение (САО). Максимальная точность в задаче определения пола составила 99,2% для набора UTKFace, минимальная – 82% для AdienceDB. Значения САО при определении возраста варьировалось от 1,07 до 10,57 в зависимости от набора данных.

Статья [123] направлена на определение пола и возраста человека по фотографии. Для решения проблемы использовалась модифицированная методами трансферного обучения модель MobileNetV2. Набор данных включал 20000 фотографий. Дополнительно проводился усложненный эксперимент по классификации фотографий людей в медицинских масках. Эффективность модели оценивалась аналогично статье [84]. Средняя точность определения пола в двух экспериментах составила 92,1%, средний показатель САО при определении возраста – 2,8 лет.

1.3 Идентификация автора сообщений в сети Интернет и продленная аутентификация пользователя социальных сетей

В связи с широкой доступностью Интернета и цифровизацией мирового сообщества многие виды деятельности частично или полностью переведены в онлайн-формат. Например, на смену почтовой переписке пришла электронная почта и социальные сети, вместо заметок в газетах и журналах люди все чаще используют

советы, публикуемые в блогах. Растущее количество авторов такого контента и возможности анонимной публикации в сети указывают на широкий спектр применения методов идентификации автора.

Публикации знаменитостей и популярных блогеров считаются авторитетными среди их подписчиков и потребителей тематического контента в данной области. В связи с высоким доверием пользователей появляются злоумышленники, стремящиеся распространить информацию от лица знаменитостей с целью получения материальной выгоды за счет призывов к выполнению определенных действий и мошенничества. Такие тексты могут содержать противозаконный контент, ложную информацию, пропаганду терроризма и экстремизма. Методы идентификации автора текста способны помочь установить личность злоумышленника.

Методы определения однородности авторского стиля сообщений [13, 65, 295] можно использовать для продленной аутентификации в социальной сетях и мессенджерах и обнаружения аномалий и необычных паттернов в потоке текстовых данных пользователей сети Интернет.

Однако надо учитывать особенности сообщений пользователей в социальных сетях. Основная из которых – длина сообщения. Как правило максимальная длина сообщения изначально ограничена настройками конкретной платформы. Так в социальной сети Twitter* она составляет 280 символов, «ВКонтакте» – до 250 символов для комментариев и до 4096 символов для сообщений, Facebook** – до 2000 символов, мессенджере Telegram – до 1024 символов и т.д. Основное предназначение этих ограничений – защита от спама [318]. Однако большую часть сообщений составляют односложные тексты длиной до трех слов, а средняя длина сообщения, как правило, не превышает пары предложений и составляет около 100 символов. Применение методов идентификации автора, в основе которых лежат статистические параметры текста, становится затруднительным.

---

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

** Деятельность организации Meta Platforms Inc, ее продукта Facebook запрещена в Российской Федерации.

Таким образом задача **идентификации автора сообщений в сети Интернет и продленная аутентификация пользователя социальных сетей на основе текстовых сообщения являются важными задачами кибербезопасности, имеющими существенные особенности в методологическом плане.**

Сводная таблица, содержащая анализ основных работ в области определения автора короткого текста, приведена в приложении А (таблица А.3).

В диссертационной работе А.А. Воробьевой [240] рассматривается идентификация автора коротких электронных сообщений и методы сравнения профилей пользователей с эталонами. Выделено несколько групп методов, первая из которых включает получение характеристик устройства, с которого осуществлялась отправка сообщения. Ко второй группе отнесена классификация по клавиатурному почерку, выделение структурных и синтаксических признаков текста и использование метода SVM. На их основе предложен метод формирования динамических стилистических профилей пользователей (рисунок 1.4), повышающий точность на 4%. Максимальная точность классификации 79,7% достигнута методом Random Forest. Предварительная подготовка данных улучшает результат на 8%.

Диссертация М.Е. Сухопарова [322] посвящена идентификации автора пользовательских отзывов, размещенных на Интернет-порталах. Предложенный автором подход фокусируется на решении существующих проблем кибербезопасности и включает модель угроз (в частности, угрозы подмены и внедрения пользователя с целью распространения ложной информации), компонентный профиль автора короткого текста, являющийся основой предложенной модели идентификатора пользователя, методики идентификации и аутентификации пользователя (рисунок 1.5) и систем защиты на их основе.

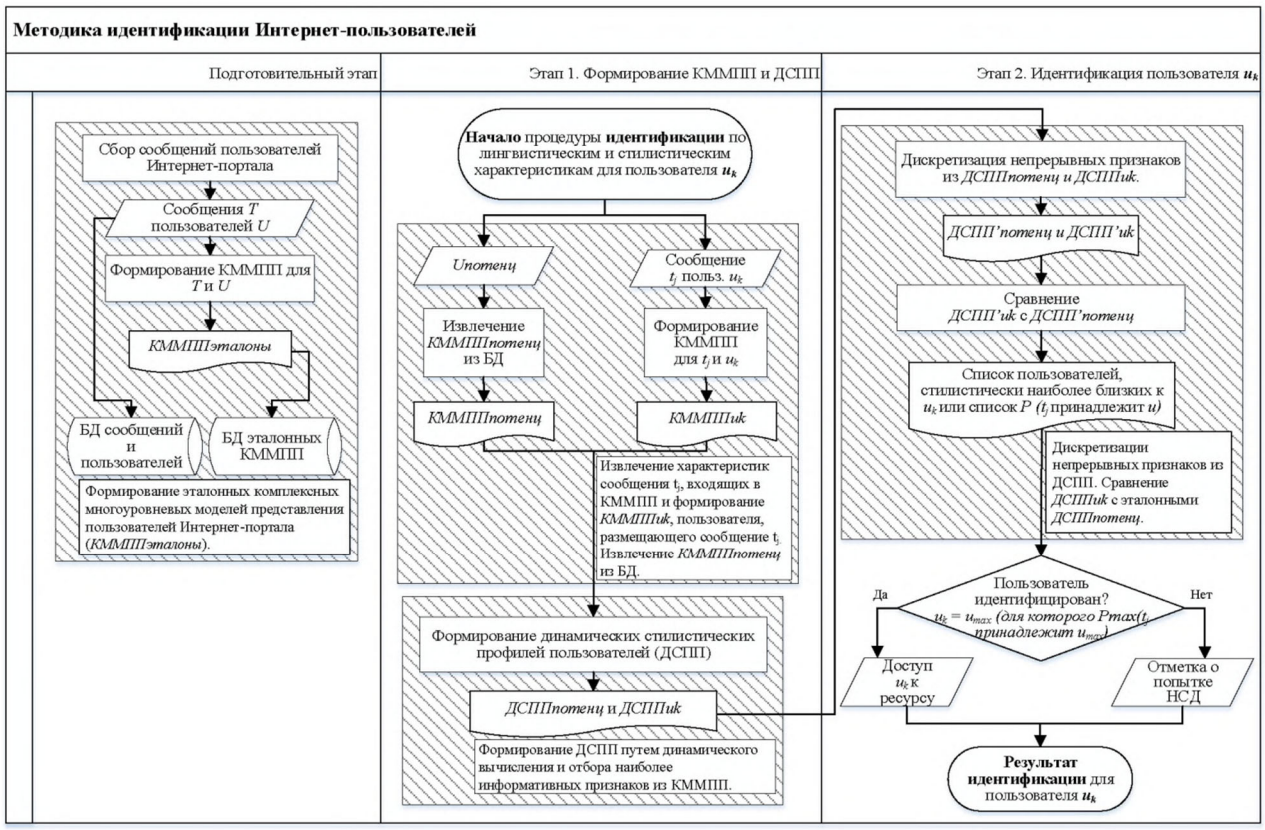


Рисунок 1.4 – Методика идентификации пользователей А.А. Воробьевой



Рисунок 1.5 – Методика идентификации пользователя портала сети Интернет М.Е. Сухопарова

Модель М.Е. Сухопарова формировалась на основе методов математической лингвистики для выделения множества признаков на графематическом, лексикографическом, морфологическом, синтаксическом и символьном уровнях. Алгоритм формирования признакового пространства отличается использованием продукционных правил, позволяющих выделять нераспознанные лингвистическими анализаторами синтаксические и шумовые словоформы. Подмножество информативных признаков было сформировано на основе энтропийного подхода. Средняя длина пользовательского отзыва составляла 8 слов. Автором также проведен частеречный анализ пользовательских текстов, на основе которого установлено, что при написании коротких текстов пользователи предпочитают использовать существительные и прилагательные, доля которых достигает 17–27% текста в зависимости от портала. Оценка предложенной методики реализована на основе вероятностного подхода и применения байесовского классификатора. Установлено, что методика Сухопарова позволяет достичь 67% вероятности для систем мониторинга состояния информационной безопасности и системы защищенности ресурсов, а также повысить на 23% вероятность идентификации пользователей порталов сети Интернет по сравнению со стандартными статическими признаками (cookie, IP-адреса), что дает возможность применить результаты выявления потенциального нарушителя ИБ в сети Интернет.

Исследование [68] вносит свой вклад в область компьютерной криминалистики, предлагая новые гибридные подходы, основанные на сочетании рекуррентных и сверточных сетей применительно к коротким текстам в соц. сетях (RNN, CNN, LSTM и гибридные модели). В качестве признаков использовались триграммы символов. В исследовании подчеркиваются проблемы анонимного Интернет-общения, которые усложняют судебно-медицинские экспертизы и судебные дела о распространении противозаконного контента. В исследовании использовались наборы данных: блог habr – тексты, связанные с информационными технологиями, «ВКонтакте» – короткие тексты на разные темы, «Эхо Москвы» – статьи. Наилучшие результаты достигаются CNN – 87% для первого набора, 59% и 53% для 50 и 100 авторов второго набора.

Статья [104] посвящена методу верификации авторства TDRLM, основанному на стилометрическом обучении. Авторы отмечают, что слова, связанные с определенными латентными тематиками, могут негативно влиять на производительность алгоритмов машинного обучения. Это связано с тем, что такие признаки больше отражают содержание текста, чем стиль автора. Для решения данной проблемы авторы разработали словарь тематических оценок, модулирующий влияние тематики на текст. Механизм на базе такого словаря позволяет корректировать векторные представления текстов и снижать влияние тематического смещения. В качестве набора данных использовался Twitter-Foursquare, точность верификации автора составила 92,47%.

Краткие выводы:

1. Комментарии и сообщения имеют небольшую длину.

2. В отличие от художественных текстов, короткие комментарии являются неформальным средством взаимодействия в сети Интернет, и при их написании не все пользователи стремятся учесть нормы русского языка и писать стилистически и орфографически верные тексты, могут использовать англицизмы, авторские неологизмы, обсценную лексику и т.д.

3. Тенденция использования постоянного набора эмодзи может рассматриваться в качестве авторского признака. Кроме того, подобные графические средства интернациональны, уточняют интонационную окраску текста и позволяют более обширно донести мысль на ограниченном объеме символов.

4. В социальных сетях предусмотрен диалоговый формат комментариев, позволяющий отмечать другого пользователя при ответе. Некоторые пользователи отвечают на комментарии именно в таком формате, другие пишут свое мнение, не отмечая других лиц.

5. Выражение мнения связано с субъективной оценкой отношения автора к контенту. Такие записи относятся к сопряженным тематическим категориям, а текст комментария содержит предметную лексику.

6. Необходимость написания комментария может быть связана и с эмоциональной составляющей, когда публикуемый контент категорически не нравится автору или, наоборот, желанием выразить положительные эмоции.

7. Некоторые пользователи могут имитировать стиль или использовать разные аккаунты. Поэтому важно проводить дополнительные проверки для повышения достоверности результатов.

Учет этих специфических черт коротких комментариев позволит разработать методы и инструменты для анализа семантики текстов в Интернете и повышения уровня кибербезопасности.

1.4 Определение автора программного кода

Определение автора программного кода [125, 259] представляет собой задачу, состоящую в установлении наиболее вероятного автора анонимной компьютерной программы на основе набора данных исходных кодов программ потенциально возможных авторов.

Решения задачи определения автора программного кода являются критически значимыми для обеспечения безопасности в цифровой среде. Это связано с тем, что подавляющее большинство технологий, а также программных систем и комплексов, упрощающих профессиональную и повседневную деятельность человека, подвержены сбоям. Подобные проблемы могут возникать по ряду разнообразных причин, например, в результате ошибок разработчиков (при проектировании, реализации и/или внедрении), неправильной эксплуатации пользователями, неполадок смежных систем. Однако наибольшую угрозу несут сбои, происходящие ввиду преднамеренного и/или злоумышленного вмешательства.

Распространенной причиной сбоев оборудования и программных комплексов является внедрение вредоносного кода путем передачи по сети вирусных программ, использования системных уязвимостей или совершения фишинговых атак. Несмотря на то, что деятельность по созданию, использованию и распространению

вредоносного программного обеспечения запрещена на законодательном уровне и закреплена в ст. 272, ст. 273, ст. 274 Уголовного Кодекса Российской Федерации [326], технические средства, обеспечивающие эффективное и своевременное установления авторства исходного кода в рамках компьютерных экспертиз, на начало 2025 года отсутствуют. Анализ исходных кодов программ на предмет авторства осуществляется специалистами в области компьютерной криминалистики вручную или с использованием малоэффективных для данной задачи средств текстового анализа.

За нарушение авторского права и интеллектуальной собственности согласно ст. 146 Уголовного Кодекса РФ [326] предусмотрена ответственность в виде штрафа, исправительных работ или лишения свободы. Высокоточные автоматизированные решения, направленные на разрешение споров об авторском праве исходных кодов программ, могут быть особенно актуальны в качестве рекомендательных систем при проведении экспертиз или для использования коммерческими организациями с целью разрешения юридических споров.

Технические экспертизы в криминалистической практике Российской Федерации, как правило, нацелены на:

1) анализ исходного кода нескольких экземпляров программного обеспечения (например, для решения споров о правах на программное обеспечение);

2) анализ исходного кода с целью выявления уязвимостей исходного кода (например, для анализа защищенности программного обеспечения или информационных систем);

3) анализ исходного кода с целью атрибутирования (выявление признаков контрафакта, компонентов иного продукта).

Ввиду отсутствия эффективных средств установления истинного автора вредоносного программного обеспечения и потенциальной сложности данной экспертизы, число преступлений в области компьютерной безопасности с каждым годом растет. В противовес этому, раскрываемость подобных преступлений снижается [275], что также подтверждает важность создания высокоточных методов решения задачи определения автора программного кода.

Еще одной проблемой, смежной с нарушением авторского права, является плагиат в образовательном процессе. Решения, позволяющие идентифицировать истинного автора программного кода, могут сделать оценку работ студентов технических специальностей высших образовательных учреждений более объективной за счет выявления в них фактов заимствования и плагиата.

Таким образом, **задача определения автора-вирусописателя представляют особую важность для кибербезопасности и компьютерной криминалистики.**

Большинство современных решений задачи определения автора программного кода базируются на анализе таких аспектов, как опыт и квалификация программиста, его персональные привычки, профессиональные приемы и непосредственно стиль написания кода. Несмотря на то, что обозначенная задача схожа с задачей определения автора естественно-языкового текста, следует учитывать, что программные коды, как объект исследования, имеют свои отличительные особенности.

Решению задачи определения автора исходного кода программы посвящено множество научных исследований. Среди трудов российских исследователей можно выделить только диссертационную работу В.Д. Стремоухова [320], посвященную установлению автора бинарного кода. Целью исследования автора является разработка гибкого и универсального метода определения авторства вредоносного кода программы с учетом особенностей реализации такого рода программ. В качестве методов решения задачи автором используются различные методы математического анализа и статистики, включая Марковские случайные процессы, теорию распознавания образов и корреляционный анализ. В рамках исследования В.Д. Стремоухов представляет несколько моделей определения авторства кода. Одной из них является модель на основе метода сжатия данных и относительной энтропии. Такой подход подразумевает объединение анонимного кода с образцами различных вредоносных кодов, написанных заранее известными авторами, и оценку того, какое из таких объединений позволяет добиться лучшей степени сжатия. Основой еще одной модели, предложенной в диссертационной работе,

является анализ матрицы переходных вероятностей Маркова. Отличительными особенностями данной модели являются независимость от ограниченной коллекции вероятных авторов, использование словаря слов вариативной длины, способность к отсечению неинформативных и/или запутывающих фрагментов кода, не отражающих стиль программиста. В качестве объектов исследования для апробирования моделей используются исполняемые файлы подсистемы win32 для операционных систем семейства Windows, а также вирусные коллекции ЗАО «Лаборатория Касперского». Вычисленные метрики качества достигают 100%, что указывает на то, что предложенные модели являются эффективными для решения задачи установления авторства вредоносного кода. Несмотря на высокие показатели точности, авторские модели имеют существенный недостаток, так как требуют единообразия языка программирования, платформы и компилятора для всех образцов исследуемого кода. Этот аспект делает предложенный метод неэффективным в ряде практических задач, связанных с определением автора программного кода в сложных случаях.

Подходы на основе математики и статистики являются популярными среди исследователей на протяжении десятков лет за счет простоты их реализации и применения, высокой скорости и низкого потребления вычислительных ресурсов. Однако такие подходы в большинстве прикладных задач уступают в эффективности современным методам на основе алгоритмов машинного и глубокого обучения.

Сводная таблица, содержащая анализ основных работ в области определения автора исходного кода программы, приведена в приложении А (таблица А.4).

Существует два основных подхода к разработке алгоритмов машинного и глубокого обучения с учетом описанных аспектов. Первый из них подразумевает **предварительное определение множества информативных признаков исследователем вручную**. Однако программист может намеренно запутать код программы или убрать из него явные авторские признаки [42, 139, 213, 216, 223]. А эксперт должен обладать глубокими знаниями парадигм, концепций и синтаксиса языков программирования и иметь богатый практический опыт. Второй подход заключается в извлечении как явных, так и неявных информативных признаков в процессе

обучения **глубоких архитектур НС**. Оба подхода имеют преимущества и недостатки, подтвержденные в ряде научных работ.

Подход Source Code Author Profile (SCAP) [83] основан на формировании авторских признаков на байтовом уровне, используя методы математической статистики для вычисления эталонных значений, формирующих профиль автора, и отклонений от этих значений для анализируемого образца. Эксперименты проводились на исходных кодах программ на языке Java. Метод достигает точности в 88%, при этом, согласно статье, может быть перенесен на другие языки программирования, а также применяться в случае, когда на одного автора приходится ограниченное число коротких экземпляров исходных кодов.

Авторы статьи [12] представили метод на основе абстрактных синтаксических деревьев (АСД) в сочетании с простой и двунаправленной LSTM. Формирование вектора признаков производилось путем их извлечения из АСД-представлений исходных кодов программистов с помощью НС. Такие представления содержат информацию о структурных элементах кода, названиях методов и функций, а также вызовах библиотек. Точность подхода при анализе набора 25 программистов Python составила 92% в случае LSTM, 96% в случае BiLSTM. При проведении эксперимента для 10 C++ разработчиков точность достигла 80 и 85%, соответственно.

Авторы [229] предлагают подход к определению авторства исходного кода на основе классификатора, обученного на комбинации признаков, извлеченных из нечетких АСД, стилометрических признаков и признаков, указывающих на дефекты авторского кода, подлежащие рефакторингу согласно парадигмам языка программирования (перегруженность кода вложенными циклами, пустыми классами, длинными комментариями и др.). В качестве классификаторов рассматривались SVM, J48, NB и KNN. Лучшая точность достигнута SVM – 75%.

R. Harang и соавторы опубликовали серию работ, направленных на определение автора исходного кода программы. Их базовый подход [44] разработан на основе сочетания RF и АСД. Подход апробирован на наборе данных соревнования Google Code Jam (GCJ), из которого были отобраны исходные коды C++ и Python разработчиков, 250 человек в каждой группе, для каждого из которых имеется

9 исходных кодов. Средняя длина исходного кода – 70 строк. В качестве признаков использовались статистические метрики, включающие число пустых строк, отступов, отношение числа отступов к длине исходного кода и др. Для отбора информативных признаков использовался энтропийный подход. Методика протестирована в условиях закрытого и открытого множеств авторов-кандидатов. Результаты закрытой атрибуции – точность 90% и 77% для Python и C++ разработчиков, соответственно. В условиях открытой атрибуции классификация проводилась для 5 классов, один из которых включал исходные коды, внедряемые на этапе тестирования и потенциально не принадлежащие ни одному из 4 авторов. В этом случае точность составила 88% и 54% для Python и C++, соответственно. Методика была доработана [43] за счет включения калибровочных кривых для анализа неполных и некомпилируемых исходных кодов. Апробация новой версии методики проводилась на собственном наборе данных, включающем 150 образцов каждого из 106 C++ разработчиков. Эксперименты проводились на выборке из 25 программистов в случае закрытой атрибуции, в открытой образцы еще 81 разработчика использовались как дополнительный класс. Авторы отмечают, что модифицированная методика способна анализировать обфусцированные исходные коды – 73% для закрытой атрибуции и не более 60% для открытой.

Исследование [34] представляет методы объяснимого искусственного интеллекта (XAI) для решения проблем, связанных с необходимостью повторного обучения при добавлении новых авторов и неинтерпретируемостью моделей. Авторы используют комбинацию CNN для извлечения признаков и метода KNN для классификации. Результаты показывают, что модель эффективно выделяет характерные особенности авторов в исходных кодах. Это позволяет не только повысить точность определения авторства, но и упростить добавление новых авторов без необходимости полного переобучения системы. Эксперименты проводятся на данных GCJ для языков C++, Java и Python. Средняя точность подходов достигает 75%.

Исследование [35] посвящено подходу к идентификации автора исходного кода, не зависящему от языка программирования. Авторы предлагают собственный подход к сбору данных, позволяющий учесть аспекты, важные для решения

практических задач. Подход предполагает использование двух алгоритмов: НС на основе пути (PbNN) и модифицированный случайный лес (PbRF). Оба алгоритма работают с представлениями кода на основе путей, строящихся для случайного фрагмента кода, и являются независимыми от языка. Точность PbNN для языка Java составила 97,9%, а точность PbRF – 98,5%.

Работа [142] выполнена в рамках соревнований AI-SOCO в рамках форума FIRE-2020, посвященного информационному поиску. Статья посвящена системе идентификации автора компилируемых исходных кодов, реализованных на языке программирования C++. В качестве алгоритмической основы системы авторы рассматривают две комбинации: TF-IDF в связке с простой НС TFIDF_NN, а также Word2Vec в связке с CNN. В качестве набора данных используются исходные коды на C++, предоставленные организаторами соревнований AI-SOCO. Общее количество исходных кодов в наборе составляет 100 тыс., где для каждого автора-программиста имеется не менее 100 образцов. Лучший результат показывает Word2Vec с CNN – точность составляет 90,86%.

Система идентификации авторства кода на основе глубокого обучения (RoPGen, рисунок 1.6) описывается в работе [228]. Исходные коды векторизуются с помощью TF-IDF. Далее они преобразуются в глубокие представления с помощью трех слоев RNN и трех полносвязных слоев. Классификация осуществляется ансамблем случайного леса. Система опробована для компилируемых языков C, C++ и интерпретируемых Java и Python. Полученная точность составляет 95%.

Работа [206] предлагает метод графа зависимости программы с глубоким обучением (PDGDL) для идентификации авторов программ. Метод содержит 3 основных этапа. Во-первых, PDG позволяет обнаруживать краткосрочные и долгосрочные зависимости в текстах программ. Во-вторых, TF-IDF обеспечивает повышение важности каждой функции PDG в исходном коде. В-третьих, метод синтетической избыточной выборки меньшинства (SMOTE) обеспечивает решение проблемы дисбаланса в классах. Наконец, алгоритм глубокого обучения извлекает особенности стиля кодирования для каждого программиста и определяет истинного автора анонимного образца. Следует отметить, что параметры глубокого обучения

подбираются авторами эвристически и тонко настраиваются под решаемую задачу. Методология оценивалась на исходных кодах 1000 авторов-программистов GCJ и содержит экземпляры на 3 языках: C++, Java, C#. Результаты авторов достигают 98% в отдельных случаях.

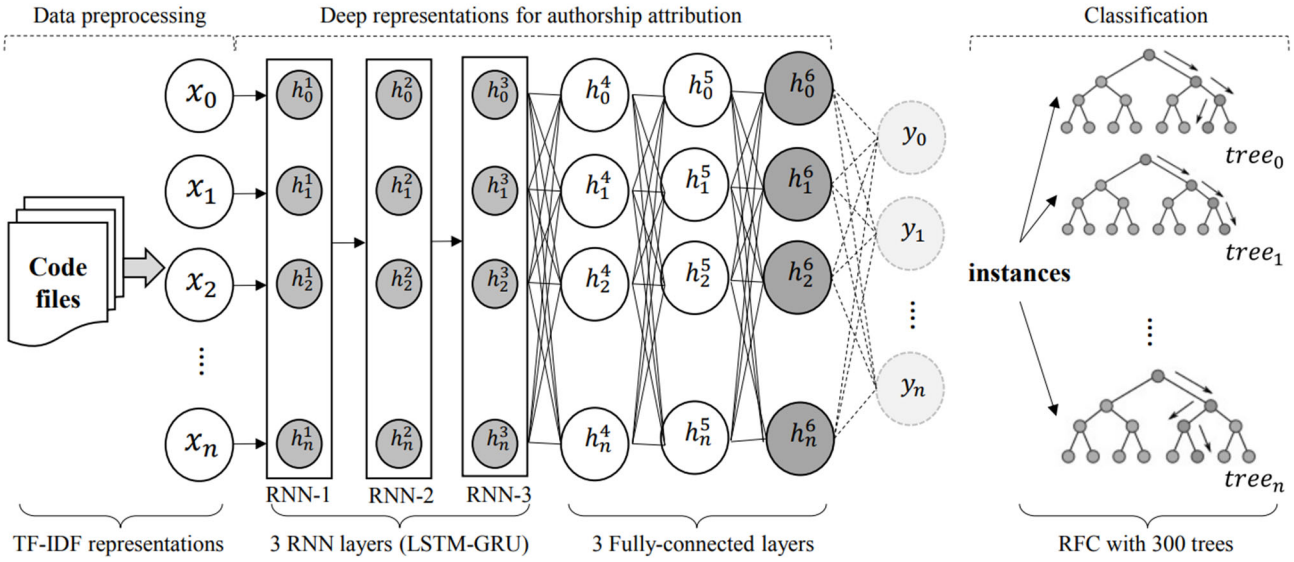


Рисунок 1.6 – Методика идентификации авторства кода программы RoPGen

Авторы работы [36] представляют ICodeNet – иерархическую НС, которая работает на уровне файла исходного кода. ICodeNet состоит из обученного ImageNet энкодера VGG, за которым следует компактная НС (вариации CNN, RNN). Отличительной особенностью метода является анализ исходного кода в формате изображения. Оценка подхода осуществляется на наборе данных с хостинга GitHub, содержащем 50 и более исходных кодов программ на языке Java для 20 авторов-программистов. Полученные результаты свидетельствуют о наибольшей эффективности сочетания моделей VGG и CNN с двумя одномерными слоями на 128 и 256 фильтров. Точность такого подхода составляет 81,33%.

Статья [14] посвящена модели для верификации авторства исходного кода на основе контрастивного обучения и трансформера в качестве энкодера (CLAVE). CLAVE принимает на вход пару исходных кодов, формирует векторы на их основе и передает их в энкодер, состоящий из 6 слоев, дообученный на 655 тыс. образцах

исходных кодов. Полученные матрицы содержат информативные стилометрические пары поданных на вход CLAVE текстов. Следующим шагом вычисляется косинусное сходство матриц. В случае, если его значение превышает заранее заданный порог, считается, что исходные коды разработаны одним и тем же программистом. Оценка качества подхода производилась на датасете GCJ, язык программирования – Python. Подход продемонстрировал 0,97 по метрике AUC и 0,93 по метрике F1. К ограничениям подхода авторы относят невозможность его применения к задаче определения авторства и ограниченную длину входных образцов – максимальный размер не может превышать 2048 токенов.

Авторы статьи [114] представляют экстрактор признаков исходного кода AuthAttLyzer-V2. В качестве алгоритмической основы выступает комбинация методов RF, градиентного бустинга, XGBoost, улучшенных за счет применения SHAP. Методы обучаются на 54 статических и динамических признаках кода, таких как комментарии, названия переменных и констант, стиль оформления блоков кода, использование памяти и др. В качестве экспериментального набора данных используются исходные коды, написанные на языке C++, с платформы Codeforces. Общее число образцов составляет 24 тыс. На каждого автора приходится по 8 образцов. В рамках эксперимента методы машинного обучения сравниваются между собой и в комбинациях. Лучший результат достигается комбинацией метода XGBoost с использованием SHAP. Точность для эталонного датасета, содержащего исходные коды 8 авторов, составляет 81,2%, полнота – 80,5%, F1 – 81,2%.

Подходы к анализу исходных кодов программ, как искусственно-языковых текстов, значительно отличаются от подходов, предназначенных для естественно-языковых. Это связано с особенностями исходных кодов программ, как объектов исследования. В отличие от художественных, любительских текстов, а также сообщений пользователей социальных сетей, исходные коды создаются в соответствии со строгими правилами, парадигмами программирования, зачастую используются известные реализации алгоритмов и шаблоны. Формальный вид текстов исходных кодов не позволяет применять подходы, обеспечивающие высокую точность определения авторства для естественных языков, к искусственным.

Для анализ искусственно-языковых текстов могут применяться как методы на основе классических алгоритмов обучения в связке с множеством признаков, так и методы на основе глубоких НС и их ансамблей. Обособленное использование классических методов не является целесообразным из-за их низкой эффективности для решения реальных задач, осложненных обфускацией, стандартами кодирования, командной разработкой и др. При выборе информативных признаков для обучения классических алгоритмов следует учитывать обозначенные ранее свойства: неизменность и невозможность умышленного искажения. Для исходных кодов такими признаками могут являться частоты определенных конструкций, пробельных символов, комментариев, уровни вложенности циклов, способ именования и оформления классов.

1.5 Определение настроения и деструктивной направленности в текстах

Тема определения настроения автора текста и тональности текста является наиболее проработанной в обработке естественного языка [6, 48, 91, 151, 163, 244]. Целью является определение эмоционального тона текста, из чего можно получить представление о различных сферах жизни человека [288]. На 2025 год общее число статей по тематике приближается к 10000.

Анализ и определение авторства текста с учетом эмоциональной составляющей имеют особую важность в контексте борьбы с экстремизмом и терроризмом. В современном информационном обществе социальные сети и онлайн-платформы стали пространством для распространения экстремистских и радикальных идей. Опасность заключается в том, что Интернет-платформы предоставляют экстремистам доступ к широкой аудитории и возможность быстрой и масштабной пропаганды своих идей. Это может воздействовать на уязвимых или подверженных влиянию лиц, включая молодежь, подстрекая их к насилию, террористическим действиям или участию в экстремистских организациях. Это создает угрозу для безопасности общества по нескольким причинам:

– радикализация и вербовка. Социальные сети и онлайн-платформы служат площадкой для вербовки новых членов в экстремистские организации путем распространения идеологии, рассказов о «героических поступках» и предложений присоединиться к их рядам;

– организация террористических актов. Социальные сети и онлайн-платформы предоставляют возможность планирования и координации террористических актов с помощью обмена информацией о тактиках, выборе целей и других деталях планируемых действий;

– легализация и финансирование экстремистских групп. Социальные сети и онлайн-платформы также могут использоваться для легализации экстремистских групп и сбора средств для финансирования их деятельности. Это может включать использование криптовалют, анонимных платежных систем или привлечение поддержки через социальные сети. Такие финансовые ресурсы позволяют экстремистам существовать, распространять свои идеи и осуществлять террористические действия.

Поэтому важно анализировать поток данных, который проходит через социальные сети. Мониторинг этой информации в определенные моменты времени и выявление лиц, имеющих целью совершение злонамеренных действий, становится актуальной практической задачей противостояния террористической угрозе и защиты государства.

Таким образом **анализ настроения автора, определения эмоциональной окраски текста, деструктивных текстов, а также текстов экстремистской направленности, запрещенных законодательством Российской Федерации, является важной задачей кибербезопасности**.

Сводная таблица, содержащая анализ основных работ в области определения тональности и деструктивной направленности текста, приведена в приложении А (таблица А.5).

1.5.1 Определение тональности текста

А.О. Корней в своей диссертационной работе [256] рассматривает методы и алгоритмы аспектного анализа тональности естественного языка (рисунок 1.7).

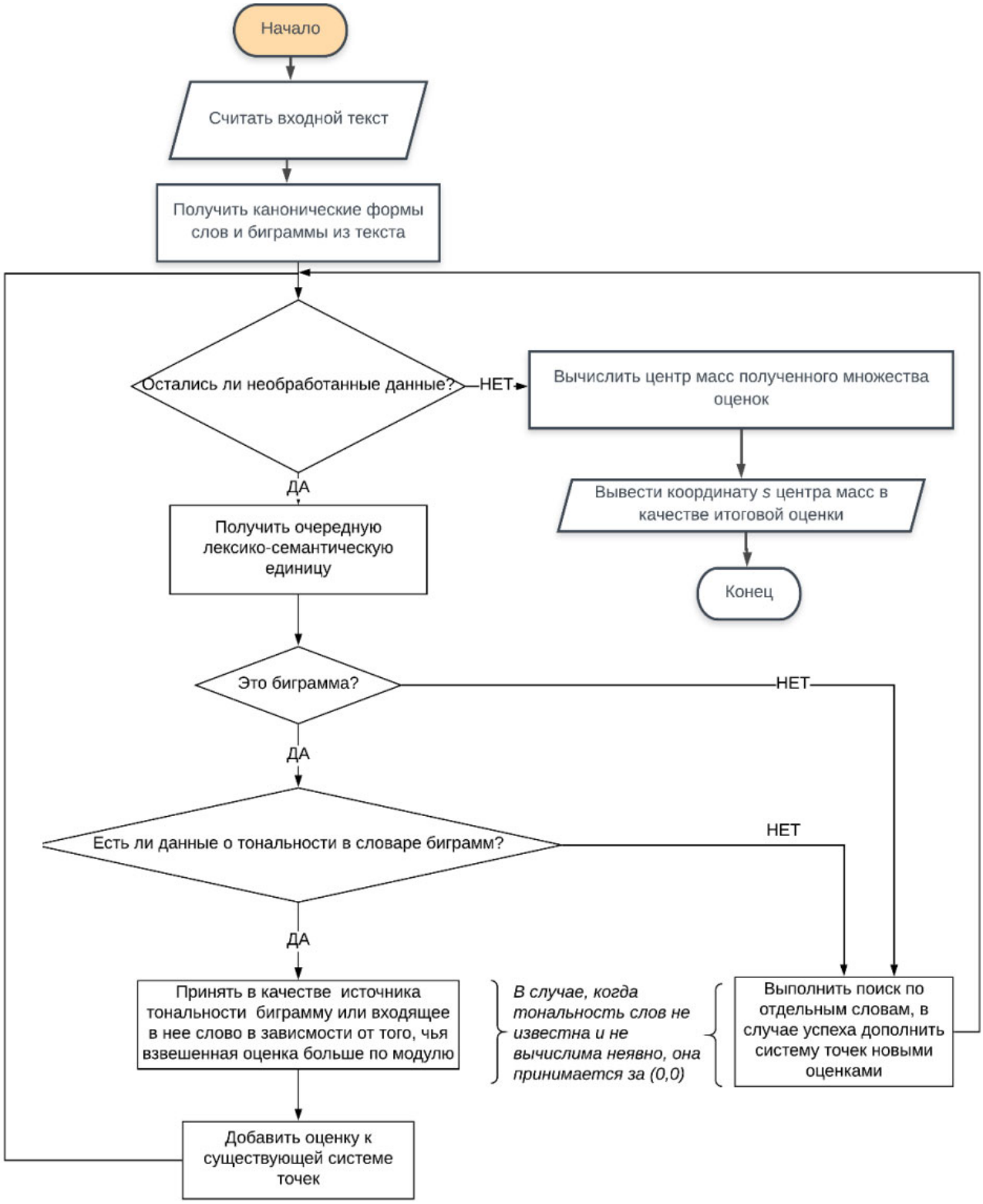


Рисунок 1.7 – Алгоритм анализа тональности на уровне семантики А.О. Корней

Разработанная модель соединяет ключевые концепции векторных представлений и структурные особенности семантических сетей и подходов на основе словарей. Согласно методике автора, вычисляются *n*-граммы первого, второго и третьего порядка на основе энциклопедии, содержащей около 7,5 тысяч словарных статей и более 2 миллионов слов. Апробация модели проводилась на корпусе, содержащем отзывы о фильмах (506 текстов) и отзывов о ресторанах (312 текстов). Для построения тематических подграфов применялся метод конденсации исходного графа на основе данных, извлеченных из обучающей выборки. При тестировании метода было обработано 314 предложений, из которых верно были определены 218 (69,5%). При внедрении разработанного графа в систему определения тональности русскоязычных постов в Twitter* достигнуто следующее: для 200 положительных и 200 отрицательных твитов точность классификации позитивных текстов – 73%, негативных – 69%.

И.А. Андреев в диссертационной работе [230] провел исследование для методов и алгоритмов обработки текстовой информации социальных сетей в задачах формирования социального портрета профиля. Данные собраны из социальных сетей и размечены по 7 эмоциям (радость, грусть, злость, удивление, отвращение, презрение, страх). Автором приводится описание оригинального подхода к сопоставлению профилей пользователей в разных социальных сетях на основе критериев: схожесть контактной информации профилей, наличие сходства лиц на фотографиях, сходства места работы и учебы, наличие схожих сообщений со страниц профиля посредством нахождения расстояния Левенштейна и использования алгоритма шинглов. Для исследования были отобраны 100 пользователей, имеющих профили в различных социальных сетях (всего 204 аккаунта). Для каждого из аккаунтов были подобраны похожие профили с помощью разработанного алгоритма. Установлено, что при использовании 566 сообщений длиной от 90 до 110 символов и несбалансированности выборки моделью BERT удалось достигнуть точности 86%. Максимальный результат для сбалансированной выборки получен с применением 2100 сообщений аналогичной длины (83%). Также автором представлен

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

алгоритм анализа эмоциональной окраски русскоязычных социальных сетей на основе BERT. Минимальная точность в экспериментах оставила 58%, а максимальная – 93% в зависимости от психологического показателя. Разработанный подход к определению психологических характеристик пользователя социальных сетей апробирован с использованием SVM с показателем точности 65%.

Ю.В. Рубцова в диссертационной работе [301] предложила подход к решению проблемы определения тональности текстов на основе автоматического извлечения словаря признаков. Корпус текстов для исследования включает авторский датасет (77 миллионов сообщений пользователей социальной сети Twitter*). Статистический анализ лексики за разные годы (2013–2018) выявил, что классификационная модель должна учитывать обогащение словарного запаса современного языка – точность анализа тональности может снизиться до 23%, если не вносить никаких изменений в модель. Предложено несколько подходов. Первая методика применяет схему взвешивания с линейной сложностью обработки, позволяющую актуализировать словарь и повторно настраивать классификатор. При этом изменение результативности классификатора на данных за 2013 год по сравнению с данными за период 2013–2018 годы оказалось ниже 1% с использованием метода мешка слов. Вторая методика включает использование тональных словарей, что помогает уменьшить различие в качестве классификации между коллекциями 2013 и 2018 года до 8,2%. Классификатор при этом демонстрирует стабильное качество на уровне 0,65–0,73. Третья методика использует модель Skip-gram и в качестве признаков применяет усредненные векторы слов из сообщений. Разница в показателях *F*-меры между 2013 и 2018 годами составила всего 0,16%.

С.В. Вычегжанин в диссертационной работе [241] разработал метод распознавания точки зрения автора текста («за», «против», «нейтрально», «противоречиво», «невозможно определить точку зрения»), представленный на рисунке 1.8. Используются авторские алгоритмы для снижения размера признакового пространства и получения множества релевантных признаков на основе индекса Джини, который оценивает степень важности признаков, классификатора SVM и процедуры

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

перекрестной проверки. Сокращение количества признаков позволило автору увеличить значение *F*-меры с 0,764 до 0,777 (на 1,3%). На полученных признаках обучается ансамбль классификаторов, включающий SVM, NB, fastText, метод KNN, дерево решений и адаптивный бустинг. Авторский метод работает на 1,5% точнее, чем модель RuBERT для корпуса сообщений из социальной сети «ВКонтакте» по темам «Вакцинация детей», «Клонирование человека» и «ЕГЭ в школе».



Рисунок 1.8 – Метод распознавания точки зрения автора текста С.В. Вычегжанина

В 2024 году было проведено соревнование PAN, направленное на мультиязычную детоксикацию текста. Предоставленные наборы сбалансированы и содержат 30000, 2700 и 720 текстов на русском, украинском и испанском, соответственно. Победителями [62] представлена система MultiParaDetox, основанная на моделях-трансформерах LLaMa и BloomZ, а также mBART и condBERT. LLaMa и BloomZ применялись для идентификации токсичных слов и выражений, модели BERT – для генерации замен выделенных фрагментов. Предобработка включала фильтрацию токсичных комментариев по ключевым словам и отбор текстов на целевом языке для каждого набора данных. Результаты оценивались по метрике Style Transfer Accuracy (STA). STA составила 95%, 69% и 45% для русского, украинского и испанского, соответственно, что превосходит результаты других участников на 2–40%.

1.5.2 Определение деструктивной и экстремистской направленности в текстах

Тональный анализ текста находит свое применение в сфере информационной безопасности [1, 181]. В работе [251] под деструктивным контентом понимают информацию, выражающую негативное отношение к личности, социальной группе, общественному классу, обществу в целом и обесценивающую, принижающую их качества и социальную значимость (пропаганда экстремизма и терроризма, критика существующего в стране порядка, разжигание межнациональной розни и пр.). Для формирования библиотеки составляется словарь терминов, определенных как в Доктрине информационной безопасности Российской Федерации, так и в других нормативно-правовых актах. Каждый термин относится к конкретной теме. Например, к теме пропаганды наркотиков отнесены слова «вещество», «доза», «наркомания» и др. Результатом работы алгоритма является процент, высчитанный по каждой теме из библиотеки деструктивного контента и определяемый как отношение числа вхождений найденных слов к общему объему текста.

Т.А. Литвинова и соавторы в своей статье [135] рассматривают применение методов анализа авторства в области онлайн-безопасности. Авторы проводят исследование на основе реальных данных российских форумов, используя различные типы характеристик *n*-грамм. В данном исследовании была проведена атрибуция авторства корпуса сообщений пользователей тематических форумов, содержащих противозаконный контент, предоставленного Azsecure-data [59]. Исследование включало анализ большого количества тем на форуме и создание набора данных, состоящего из сообщений 20 активных участников, для проведения экспериментов по атрибуции авторства в рамках одной и разных тематик. Отдельной задачей стал отбор 20 активных участников, поскольку исходный датасет содержит более 1000 авторов и 699000 сообщений, многие из которых посвящены бытовым вопросам и не относятся к противозаконному контенту. В качестве модели использовали LinearSVC. Полученная точность составила 48,8% для 20 авторов.

Исследовательская группа во главе с В.А. Минаевым и А.В. Симоновым [270, 272] проводила анализ деструктивного и экстремистского контента в социальных сетях. Под этим они подразумевают контент, который включает призывы к вражде на основе национальности и религии, распространение информации о наркотиках, порнографические материалы, а также пропаганду терроризма и экстремизма. В первой статье рассмотрены модели для классификации текста и методы его предобработки для выявления деструктивных элементов в социальной сети «ВКонтакте». Данные были разделены на три категории: нейтральные публикации, публикации об исламе без экстремистской направленности и публикации о радикальном исламе. Применялись основные техники векторизации текста, такие как BOW, TF-IDF и Word2vec. В результате, интеграция алгоритма BOW, метода главных компонент для сокращения признакового пространства текстов, а также использование моделей LR или RF показали точность 97% в распознавании деструктивного контента. Однако в работе не уточняется методика, по которой определялись радикальные публикации, и не объясняется процесс разделения данных на выборки. Во второй статье авторов обсуждается целесообразность использования моделей НС, в частности BERT, но конкретные методы их реализации и применения не рассматриваются.

И.В. Машечкин и др. опубликовали работы [267, 269], посвященные категоризации текстов экстремистской направленности на Интернет-ресурсах. Метод автоматического аннотирования и выделения ключевых слов для поиска информации экстремистского содержания в потоках текстовых сообщений апробирован на датасете KavkazChat* [59]. Поскольку не все тексты данного набора относятся к экстремистской тематике, первоначально был применен алгоритм иерархической кластеризации для формирования 10 кластеров (включая обычные тематики: кулинария, автомобили и т.д.; и потенциально опасные: терроризм, война, оружие, радикальная политика и религия). Установлено, что в большинстве текстов присутствует политический контекст. Далее применялся метод выделения ключевых слов

* Признан экстремистским в Российской Федерации.

с удалением информационного шума, основанный на использовании неотрицательной матричной факторизации для матрицы термов веток текстовых сообщений. На основе кластеров были извлечены ключевые слова, которые были применены к задаче противодействия терроризму и экстремизму в работе [269]. В этом подходе вместо традиционного поискового запроса используется документ-образец, а целью поиска является нахождение документов и сообщений релевантных этому образцу. Из документа-образца выделяются ключевые слова на основе ортонормированной неотрицательной матричной факторизации и представления слов в виде *n*-грамм. На основе выявленных ключевых слов формируются поисковые запросы. Возвращенные результаты, содержащие шумы и ошибки, подвергаются ранжированию с использованием оригинального метода. На основе предложенного метода фильтрации набор данных KavkazChat* был обогащен комментариями пользователей сети «ВКонтакте», в которых встречаются отобранные ключевые слова. Классификация текстов на три группы (опасные, неопасные, неизвестные) была произведена с помощью LR с L2 регуляризацией, дерева решений типа CART, RF и ансамблей деревьев решений на основе градиентного бустинга XGBoost и LightGBM. Параметры алгоритмов подбирались с помощью стратифицированной кросс-валидации. Установлено, что при разбиении набора на 10 кластеров с различными деструктивными тематиками точность варьируется от 86 до 94%. Лучший результат получен с помощью XGBoost.

Статья А.А. Гончарова [243] ориентирована на проведение автоматизированного анализа текстов экстремистской направленности с целью сокращения временных затрат на экспертизу текста. В частности, авторы выделяют такие темы как «религиозный экстремизм», «исламский экстремизм», и «исламский террористический интернационал». В качестве алгоритмов машинного обучения были рассмотрены: SVM, KNN, NB и RNN. Набор данных содержит неструктурированные тексты с заранее известными классами: классический ислам, радикальный ислам, нейтральные тексты. Сбор данных для обучающей выборки осуществлялся совместно с сотрудниками Центра по борьбе с экстремизмом. За основу поиска брались материалы, внесенные в список запрещенных. Так же различные тексты

экстремистской направленности брались из журналов таких как «Румия» и «Истоки», данные журналы занимаются пропагандой идей Исламского Государства (итого 300 текстов). Информация для классов «Классический Ислам» и «Нейтральная тематика» извлекалась из открытых сайтов сети Интернет (1800 статей). Все тексты были приведены к нижнему регистру и очищены от знаков препинания. Также проводилась векторизация текстов, при этом метод авторами не уточняется. Точность методов составила 87%, 53%, 53% и 45% для RNN, KNN, наивного байесовского классификатора и SVM, соответственно.

L. Ranaldi и соавторы в своей работе [181] исследуют установление авторства текста текстов англоязычного форума, посвященного обсуждению вопросов, связанных с противозаконными организациями. Набор данных состоит из 91874 сообщений с 13995 обсуждениями и 2082 активными пользователями. Всего были сформированы три набора данных: в первом наборе данных были отобраны три пользователя с большим количеством постов и более высокими баллами радикальности. Во втором наборе данных – пять пользователей с теми же критериями, а в третьем наборе – десять пользователей по тем же критериям. Каждый набор данных содержит 1400 постов на пользователя. Для каждого текста рассчитывается TF-IDF представление, а также выделяются лексические и стилистические признаки с помощью kernel-inspired encoder with recursive mechanism for interpretable trees (KERMIT) [227], которые затем передаются на вход SVM. SVM с предварительно сформированным признаковым пространством показал результаты для 3, 5 и 10 авторов равные 85%, 75% и 59%.

В статье [149] представлен алгоритм для обнаружения экстремистских текстов, включающий предобработку, фильтрацию с использованием словарей и классификацию текстов с помощью NB и SVM. Авторы создали мультиязычный словарь экстремистской лексики, разделенный на четыре уровня: умеренный, нейтральный, низкий экстремизм и высокий экстремизм. Экспериментальная выборка включала 190 экстремистских постов и 820 подозрительных текстов, разделенными на тренировочный и тестовый наборы в соотношении 9 к 1. Точность классификации на тестовой выборке составила чуть более 84%.

Авторы статьи [151] собрали собственный корпус текстов, состоящий из экстремистских и неэкстремистских постов «ВКонтакте», включающий 3000 слов для экстремистских и 15000 слов для неэкстремистских постов. Для бинарной классификации использовались модели Gradient Boosting с Word2Vec и TF-IDF, а также Random Forest с Word2Vec и TF-IDF. Основные признаки включали лексические (мешок слов, *n*-граммы, частота слов) и синтаксические признаки, а также анализ тональности. Лучшие результаты показал Gradient Boosting с Word2Vec, достигнув точности 89%.

В статье [118] использовали набор данных Dark Web, собранный Artificial Intelligence Lab для автоматической классификации текстов на радикальные и нерадикальные. Для классификации использовались методы NB, KNN, SVM, DT, LR, Dynamic Query Expansion. Признаки текста включали лексические, синтаксические и тональные характеристики, такие как частота использования слов, анализ тональности текста. Очистка текста включала удаление ссылок, эмодзи, исправление опечаток и нормализацию текста. Точность достигала 88,11%, при использовании NB.

Авторами работы [3] предложено решение проблемы классификации текстовых сообщений с информацией о подготовке террористических атак. Методика включает комбинированное использование статистических признаков и ключевых атрибутов из Global Terrorism Database. Набор данных состоит из различных записей о террористических атаках с ключевыми атрибутами, включая тип атаки, город, страну, террористическую организацию, тип оружия и был ли запрошен выкуп. Методика основана на модели DistilBERT, включающей 6 слоев Transformer. Датасет был разделен на семь классов: убийства, вооруженные нападения, взрывы, угоны и другие. Точность классификации составила 97%.

В работе [268] показано, что выявление авторов с высоким уровнем агрессивности как особенности личности позволяет эффективнее распознавать речевую враждебность, троллинг, кибербуллинг, различные формы манипулирования и подстрекательство к розни. Эмоциональная составляющая оценивается как на уровне всего текста, так и на уровне отдельных высказываний. Для экспериментов использовались сочинения 487 испытуемых. Признаки были расширены путем

модернизации группы семантических характеристик. Каждому автору присваивалась категория агрессивности – физическая, гневная, враждебная или нейтральная. Таким образом задача предсказания значения целевого признака сводилась к двухэтапной классификации, где в первом случае целевой меткой был идентификатор автора, а во втором – агрессивности. Точность определения автора составила 62,1%, а для определения агрессии лучший результат 59,1% был получен при использовании семантических и частотных признаков и SVM.

Поиск решений проблемы выявления текстовых признаков, специфичных для авторов с определенными психологическими отклонениями, затронут в продолжении работы [268] тех же авторов [122]. Метод семантического анализа, предложенный авторами, опирается на работу Н.К. Онипенко [279], в частности, на синтаксемный анализ и аппарат неоднородных семантических сетей. Предварительно был составлен словарь предикатов, содержащий 2700 элементов (глаголы, деепричастия, причастия, семантические отношения и семантические связи). В качестве анализатора использовался PLATIn, преимуществом которого является поддержка анализа веб-страниц. Оценка текста проводилась по параметрам рациональности, эмоциональности, экспрессии и социального стресса, для каждого из которых создавались отдельные словари. В сумме словари насчитывают более 51000 элементов. Помимо семантических, использовались частотные распределения 14 частей речи. В ходе проведенного исследования приняли участие 487 человек, среди которых дети, студенты и взрослые. Каждому участнику было необходимо написать сочинение на тему «Я, другие, мир» объемом в одну страницу и ответить на десять тестовых вопросов. При обработке данных использовались алгоритмы бинарной классификации: SVM и RF. Полученные результаты позволяют сделать вывод о важности применения частотных распределений (это позволило увеличить точность классификации на 10%), поскольку в ряде случаев отклонение употребления местоимений в текстах позволяет выявить нарциссические черты личности.

Исследование [160], посвящено определению контента с признаками агрессии в коротких русскоязычных текстах из социальной сети «ВКонтакте» за 2015–2019 гг. Под агрессивным контентом, выраженным в форме текста, авторы

понимают целенаправленное использование оскорбительной лексики в адрес индивида, в том числе нецензурной, открыто унижающей честь и достоинство получателя сообщения. В методе используется модель мешка слов и *n*-граммы и RNN для классификации сообщения как нейтрального или агрессивного. Выбор RNN обоснован ее способностью к анализу контекста сообщений. Согласно результатам, максимальная точность 85% достигается при 10 эпохах обучения НС и использовании 16000 коротких сообщений.

Статья А.О. Исхаковой и Р.В. Мещерякова [110] является логическим продолжением работы [160] и направлена на проведение научных исследований в области выявления признаков деструктивности в текстах, опубликованных в социальных сетях, с целью реализации социокиберфизических систем разнородного Интернет-контента: текстов, изображений, видео. Датасет сформирован на основе текстов пользователей русскоязычного развлекательного портала Pikabu. Все тексты были размечены по тональности (3071 негативных, 7023 нейтральных, 2239 позитивных). При проведении предварительной обработки были удалены специальные символы, стоп-слова, ссылки при ответах другим пользователям, а также все тексты приведены к нижнему регистру. Для классификации выбраны fastText, LR и MultinomialNB и метод представления текста в векторном виде BOW. Результаты экспериментов показывают, что с точки зрения точности лучшим методом оказался MultinomialNB (64%), а с точки зрения времени, затраченного на обработку текста – fastText (1,52 секунды). Достаточно низкие показатели точности обусловлены малым размером текстового корпуса, а также факторами, связанными с квалификацией экспертов.

В диссертации Ю.В. Давыдовой [245] исследуются методы мониторинга контента в социальных сетях с целью реализации превентивных мер пропаганды криминализации. Предложены техника специализированного текстового поиска на основе динамического программирования и деревьев решений, новый подход, основанный на семантическом анализе для определения жаргонизмов и неологизмов, тематической лексики пропагандистов криминала, гибридная модель, сочетающая лингвистические правила и статистические данные для присвоения веса различным

типам текстовых ошибок. Данные для исследования включали федеральные и региональные базы данных правоохранительных органов и платформы социальных сетей. Был разработан прототип программного обеспечения, который апробирован в реальных сценариях для мониторинга и анализа социальных сетей на предмет незаконной деятельности. Предложенный подход продемонстрировал точность 95% при обнаружении противозаконной информации.

Несмотря на то, что тема тонального анализа наиболее широко представлена в публикациях и в готовых программных продуктах, результаты исследований для прикладных задач кибербезопасности показывают низкую точность. Еще одной проблемой является отсутствие в открытом доступе больших корпусов, содержащих тексты деструктивной и экстремистской направленности.

1.6 Анонимизация, имитация и генеративные модели как методы атаки на методы идентификации автора

Существуют эффективные программные решения, позволяющие учитывать различные лингвостатистические параметры текста, изменять их и дополнять тексты без потери смысла [186, 258]. Статистические параметры текста можно изменить и вручную. Таким образом можно осуществлять атаки на методы идентификации автора текста (рисунок 1.9).

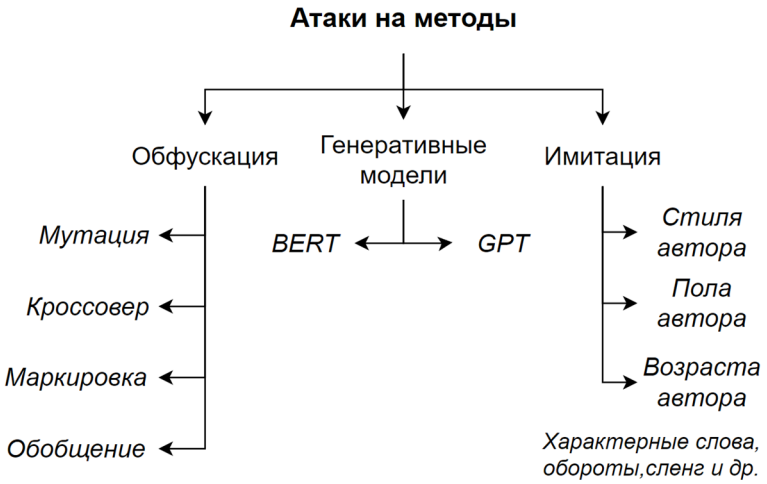


Рисунок 1.9 – Атаки на методы идентификации автора

Обфускация представляет собой процесс намеренного сокрытия данных с целью обезличивания автора. В качестве таких данных в задаче атрибуции текста выступают признаки, характеризующие стиль письма человека, его поведенческую биометрию: часто используемые слова и речевые обороты, структурные особенности построения предложений и т.п. Извлеченные из текста информативные признаки в дальнейшем подвергаются модификации, направленной на устранение уникальных черт с сохранением ключевого смысла. В контексте естественного языка кроссовер может означать перемешивание частей текста между двумя или более предложениями или абзацами. Мутация может включать замену слов синонимами, перестановку слов в предложении, изменение грамматических конструкций и т.д. Обобщение в тексте может включать замену конкретных деталей на более общие описания. Маркировка может включать добавление бесполезных предложений или других элементов, которые отвлекают внимание от основной идеи текста.

Имитация авторского стиля представляет собой процесс создания текста, копирующего стиль письма или поведенческие особенности определенного автора, другого пола, другого возраста. В контексте машинного обучения, это может предполагать также использование генеративных алгоритмов для создания текстов, которые могут быть ошибочно приняты за тексты определенного автора. Такой процесс может быть полезен в задачах повышения устойчивости методов атрибуции текста, анонимизации данных или создания искусственных текстовых данных для обучения моделей машинного обучения [7].

Достижения в области больших языковых **генеративных моделей** позволили создать мощные инструменты, такие как ChatGPT [47], Jasper [112], Bard [27], ruGPT3 [107], Yandex GPT [221] для генерации компьютером человекоподобного текста. История Александра Жадана [323], защитившего диплом, написанный с помощью ChatGPT, и его опыт с функцией системы «Антиплагиат» [231], которая идентифицирует текст, сгенерированный ChatGPT, поднимает важные вопросы о будущем академических работ. Жадан, успешно обойдя систему, задается вопросом о ценности и необходимости дипломных работ в эпоху, когда создать оригинальный текст с помощью ИИ стало проще. Затем система «Антиплагиат»

добавила новый функционал [329], идентифицирующий тексты, написанные нейросетью. Однако Жадану в итоге удалось обойти и его, повысив оригинальность полностью искусственного текста до 94,89%.

Однако проблема подобных сгенерированных статей была известна еще раньше. Так C. Labbé [126] еще в 2010 году сгенерировал более 100 статей и подал их в различные журналы под псевдонимом. Статьи успешно прошли этап рецензирования и были опубликованы, а *h*-индекс цитируемости вымышленного автора достиг значения 94, что говорит о серьезной проблеме в академической среде.

Актуальной задачей кибербезопасности становится проблема создания и усовершенствования методик, учитывающих способы сокрытия авторского стиля (обфускация) и имитации авторского стиля, а также генеративных моделей, позволяющих автоматически генерировать тексты на основе глубоких моделей, обученных на больших текстовых корпусах (GPT). В связи с этим возникает необходимость в проведении дополнительных исследований, направленных на оценку эффективности и устойчивости методов определения авторства текста к такого рода атакам.

Сводная таблица, содержащая анализ основных работ в области атак на методы идентификации автора, приведена в приложении А (таблица А.6).

1.6.1 Обфускация

Вопрос анонимизации и обфускации естественно-языковых текстов часто поднимается в трудах иностранных исследователей [177]. Некоторые предлагаемые подходы демонстрируют положительные результаты и заслуживают особого внимания. Следует отметить, что методик для анонимизации естественно-языкового теста на русском языке, насколько известно автору, нет.

Авторы исследования [140] считают, что существующие методы анонимизации не приспособлены для длинных текстов. Для устранения этих недостатков предложена система «Mutant-X» на основе генетического алгоритма. Система

позволяет автоматически анонимизировать текст, чтобы успешно избежать атрибуции, сохраняя семантику текста схожей с оригинальным текстом. В частности, «Mutant-X» последовательно вносит изменения в текст, используя методы мутации и кроссовера, руководствуясь при этом функцией приспособленности, которая учитывает вероятность атрибуции и семантическую релевантность. Хотя «Mutant-X» требует знания модели атрибуции (противника), представленной «черным ящиком», он не использует дополнительных обучающих данных и работает с документами любой длины. Оценка происходила на корпусе PAN-2016. Результаты показывают, что «Mutant-X» может снизить точность современных методов установления авторства на 64%, сохраняя при этом семантику. Недостатком является то, что предложенная авторами методика, предназначена для ограниченного числа моделей определения авторства.

Суть работы [117] заключается в поиске способов уменьшения эффективности стандартных методов атрибуции авторства. Авторы внедряют два уровня анонимизации: поверхностный и глубокий. Были изучены возможности создания инструмента для поддержки анонимизации документа, исходя из предположения, что автор готов приложить разумные усилия в этом процессе, например, проверить орфографию и грамматику. Для исследования был выбран стандартный набор данных – The Federalist Papers. Для каждого набора признаков был использован SVM. В результате установлено, что внесение изменений в документ с целью увеличения или уменьшения частоты употребления того или иного термина обязательно повлияет на частоты других терминов и, таким образом, повлияет на стилометрию документа. Недостатком подхода является то, что он применялся в первую очередь к авторам, которые имеют достаточно большой корпус. Простые SVM и низкоразмерные модели (такие как SVM03) менее устойчивы к попыткам «затуманивания», чем метод Коппеля и подход Шлера. Создание классификаторов с минимальным количеством признаков создает более восприимчивую даже к упрощенным попыткам затуманивания модель. Точность методики составила 83%.

Авторы работы [26] рассматривают анонимизацию текста, как способ защиты личности, а не персональных данных. Их цель – маскировка авторского

стиля письма без потери смысловой нагрузки информации. Это первая работа, в которой анонимизации авторского текста представлена для открытого набора кандидатов. В реализации применялась Generative Adversarial Network (GAN), включающая генератор на основе Гауссовской случайной модели и дискриминатор для различения оригинальных и анонимизированных образцов на основе Сиамской НС. Таким образом, задача преобразовывалась в минимаксную, где генератор максимизирует разницу между исходным и целевым образцом, а дискриминатор пытается ее минимизировать и выделить сходства. При предварительной обработке текста авторы учитывали классический подход формирования признаков стиля письма на основе стилометрии (частот *n*-грамм слов и символов) и подбирали комбинацию, которая являлась бы информативной для определения автора, была максимально нейтральной и не включала специфических признаков. Поэтому авторы фокусировались на выделении признаков на уровне слов и *n*-грамм, а не предложений, а анонимизацию проводили на уровне предложений, но не абзаца и текста в целом для сохранения семантики. Реализация GAN выбрана как Wasserstein GAN with Gradient Penalty (WGAN-GP) с добавлением сверточных слоев. Эксперименты проводились на выборке из набора Reuters C50 dataset (отобрано 2500 статей 50 авторов). Для оценки качестве сгенерированного контента использовали метрики Bilingual Evaluation Understudy (BLEU), BERTScore, а также собственную метрику Anonymity, рассчитываемую как разницу точности при внедрении сгенерированных образцов при определении авторства на этапе тестирования открытой атрибуции. BLEU – 0,61, BERTScore, – 0,93, Anonymity – 0,19.

1.6.2 Имитация

Единственной работой для русскоязычных текстов, где проводится имитация авторского стиля, является диссертация А.Г. Сбоева [305]. В ней получена оценка влияния на результат намеренных искажений авторами своего возраста и пола.

Результаты показывают, что подобная атака очень эффективна, а точность распознавания попытки искажения составляет менее 50%.

Панов и др. [150] предлагает метод анонимизации английского текста для сокрытия конфиденциальной информации и изменения авторского стиля, который учитывает особенности социальных сетей. Метод основан на BERT для генерации семантических эмбеддингов с целью синонимизации без потери смысловой нагрузки. Эксперименты проводились для 10 авторов, каждый из которых создал 1000 образцов. Авторы заявляют, что точность определения автора текста без атаки анонимизированными образцами составила 77%, с добавлением образцов, созданных по методике, точность менее 5% для того же количества авторов.

Определение изменения авторского стиля ежегодно является одной из задач, предлагаемых участникам соревнования PAN. При проведении соревнования 2023 года были выдвинуты следующие требования: участникам нужно было для заданного текста найти все позиции изменения стиля письма на уровне абзаца. Предоставлены датасеты трех уровней сложности: легкий (абзацы документа охватывают различные темы, что позволяет использовать информацию о темах для обнаружения изменений авторства), средний (разнообразие тем в документе невелико, что заставляет участников ориентировать подходы на стиль письма), сложный (все абзацы в документе посвящены одной теме). Все тексты написаны на английском языке и могут содержать произвольное количество изменений стиля. Однако изменения стиля могут происходить только между абзацами. Каждый датасет содержит 6000 документов, созданных от 2 до 4 авторами в зависимости от набора данных. Команде под руководством A. Hashemi [97] удалось победить в категориях легкого и среднего наборов благодаря использованию предварительно обученных языковых моделей BERT, RoBERTa и ELECTRA, объединенных со слоем бинарной классификации. Результат – значения *F*-меры 0,98 и 0,84 для легкого и среднего наборов. С жестким набором данных удалось добиться лучшего результата команде Z. Ye [225]. Их путь к решению проблемы кроется в применении предварительно обученной большой языковой модели DeBERTaV3. Для классификации стилевых изменений используется контрастивное обучение. Результат – 0,82.

Авторы статьи [54] проверяют гипотезу об улучшении качества верификации авторства путем добавления синтетических примеров, которые имитируют стиль конкретного автора. В качестве классификаторов, обучаемых с использованием синтетических данных, они рассматривали SVM, RNN и CNN модели. В качестве экспериментальных данных использовались 5 наборов данных – TweepFake, Extended Brennan-Greenstadt (EBG), Riddell-Juola, PAN11 и Victoria. В рамках эксперимента оценивалось качество верификации моделями до и после добавления в датасет искусственно-сгенерированных при помощи модели семейства GPT образцов. Лучший результат был получен CNN, обученной на датасете с искусственно-сгенерированными образцами, и составил 81% для датасета EBG. В случае с другими наборами данных влияние выполненной аугментации оказалось негативным.

Авторы работы [95] провели многостороннюю оценку трех современных методов имитации стиля письма и показали, что все они не справляются с задачей сохранения семантики.

1.6.3 Генеративные модели

Диссертационная работа А.О. Исхаковой [252] посвящена идентификации искусственно-сгенерированных текстов. Исследование включало проверку на естественных, созданных с использованием цепей Маркова, созданных методом синонимизации текстах. Современные глубокие генеративные модели не использовались. Применение метода на основе расстояния Махаланобиса показало, что в 93% случаев можно корректно классифицировать тексты как естественные или искусственные, с ошибками первого и второго рода, не превышающими 4% и 3% соответственно.

Авторами статьи [195] рассмотрена проблема разграничения естественных и созданных генеративной моделью текстов. При проведении исследования авторы использовали эссе 126 людей, являющихся носителями английского языка. Объем каждого эссе составляет 250–300 слов. Еще 126 текстов были сгенерированы

с использованием ChatGPT. Каждый текст был подвергнут минимальной предварительной обработке, состоящей в токенизации, приведении текста к нижнему регистру, стеммингу и удалению стоп-слов. Для векторизации обработанных текстов использовался TF-IDF. Авторы приводят список выбранных ими характеристик: 120 символьных признаков – униграммы, биграммы и триграммы, процент цифр в тексте, 40 частот специальных символов, 60 признаков уровня слов – количество слов, среднее по тексту и абзацу, биграммы и триграммы слов, 60 признаков на основе частот частей речи и 2 признака уровня предложений – их количество и средняя длина по тексту. Для оценки информативности признаков использовался метод SHapley Additive exPlanations (SHAP), основанный на теории игр. Для принятия решения выбран классификатор XGBoost. Точность определения искусственного текста составила 93–95%.

Статья [200] описывает подход к распознаванию сгенерированных текстов на русском языке с использованием BERT-моделей. Задача заключалась в определении, сгенерированы ли тексты, и какой именно моделью. Методы включали трансферное обучение с использованием предобученных моделей RuRoBERTa, RuBERT, RuGPT3 и RuGPT2, при этом основная модель была дообучена с помощью Byte-level Byte-Pair Encoding токенизации. Классификация проводилась путем токенизации текстов и использования предобученных моделей, дообученных для конкретной задачи, с применением слоя классификации на выходе, имеющего 14 нейронов, соответствующих 14 классам (различные модели генерации текста и класс «человек»). Набор данных состоял из 215110 текстов. Лучшая точность, достигнутая моделью RuRoBERTa, составила 65%.

В работе [9] обращено внимание на проблему выявления текстов, созданных ChatGPT в академической среде. Использовались аннотации 500 научных статей на английском, созданных человеком, и столько же сгенерированных с использованием ИИ. Автор сравнивает производительность специальных инструментов обнаружения ИИ и классификаторов машинного обучения, включая SVM, LR и RF, а также исследует влияние пользовательских инструкций на качество текста. Для анализа текста использовались такие метрики, как стилометрия, настроение,

схожесть текста, читабельность и релевантность. Классификаторы машинного обучения, обученные на DistilBERT-эмбеддингах, показали результат *F1*-меры 99% для SVM и LR, и 96% для RF, превзойдя показатели инструментов обнаружения ИИ, который набрал 64-83%. Было обнаружено, что подробные инструкции к ChatGPT улучшают сходство с оригинальными текстами и снижают эффективность инструментов обнаружения ИИ-контента [45, 205].

Авторы статьи [45] исследовали инструменты обнаружения контента, созданного ИИ для текстов, сгенерированных ChatGPT [46]. Затем ответы, сгенерированные ChatGPT, были переведены на другие языки. Все сгенерированные тексты проверялись детекторами ИИ-контента. По мнению авторов, все пять детекторов ИИ-контента, еще не готовы точно и убедительно отделять текст, созданный ИИ, от естественных текстов.

1.6.4 Атаки на методы идентификации автора программного кода

Атаки для данного вида текста делятся на две группы: имитация стиля и обфускация.

Обфускация кода – это процесс изменения исходного и/или исполняемого кода программы таким образом, чтобы он стал сложным для понимания и анализа, но сохранил работоспособность и функциональность [128, 179, 199]. Следует отметить, что обфускация может применяться как в качестве умышленной атаки на метод, так и в качестве инструмента защиты программного обеспечения от несанкционированного доступа и копирования, поэтому теме обфускации посвящено множество научных работ и исследований.

Фундаментальное исследование, посвященное обфускации – кандидатская работа Д.Н. Буинцева [237]. Цель данного исследования состоит в разработке метода обеспечения информационной безопасности программного продукта вне доверенной вычислительной среды (рисунок 1.10). Все методы апробируются на исходных кодах программ с целью их последующей обфускации и предполагают

такие запутывания, как преобразование форматирования, преобразование структур данных, преобразование графа потока управления и превентивные преобразования. Автором выдвигаются требования, которым должна отвечать обфусцированная программа. Среди них обязательное маскирование факта запутывания, нерегулярность преобразований и устойчивость. Принцип работы метода заключается в выделении базовых блоков исходного кода программы и построении графа потока управления, разрушении структуры графа потока управления, включая различные преобразования (введение массива переменных и их переприсваивание, введение «мертвого кода»). Согласно проведенным экспериментам и полученным результатам, предложенный Д.Н. Буинцевым метод обеспечивает высокую степень скрытности деталей реализации программных средств с помощью запутывающих преобразований.

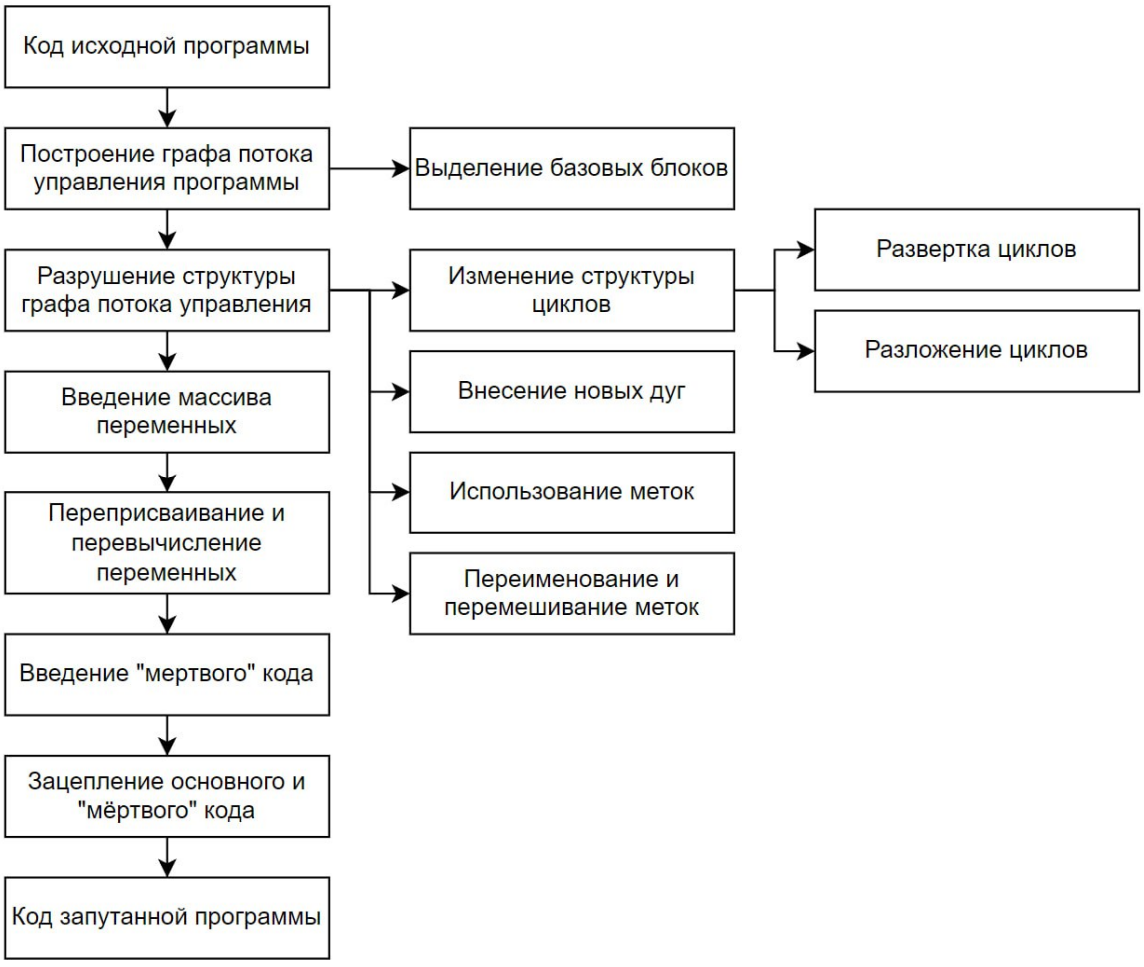


Рисунок 1.10 – Метод защиты программных средств
на основе запутывающих преобразований Д.Н. Буинцева

Еще одна кандидатская диссертация, заслуживающая отдельного внимания, представлена Е.М. Кутузовым в 2016 году [262]. Целью данной работы является создание методики обфускации программного кода и повышение защищенности информации на ее основе. В качестве экспериментальных данных используется язык программирования Java. Полученные результаты подтверждают эффективность методики для языка Java и позволяют повысить уровень защищенности исходных кодов.

Теме обфускации уделяется большое внимание и в научных статьях исследователей, занимающихся повышением качества и удобства написания кода, а также повышением степени защиты программного обеспечения. Соответственно, с каждым годом количество методов и инструментов для эффективной обфускации программного кода растет, что становится проблемой исследователей, занимающихся разработкой подходов к определению авторства.

Работа [5] посвящена целевым и нецелевым атакам на методы определения автора программного кода. Целью является выявление потерь методов в эффективности при наличии и отсутствии факта обфускации. В качестве набора данных используется GCJ с образцами кодов 200 программистов. Всего в эксперименте задействовано 6 методов, включая вариации RNN, CNN и методов стилометрии. Рассмотренные методы демонстрируют уязвимость перед нецелевыми атаками, уровень успешности которых превышает 98,5%. Они снижают эффективность методов определения авторства на 13% и более. Целевые атаки подразумевают намеренную имитацию авторского стиля с целью подмены авторства. Успешность таких атак варьируется от 66% до 88% для различных методов определения авторства.

Применение обфускации может оказать критическое влияние на точность методики идентификации автора по причине искажения статистических характеристик и информативных признаков, определяющих стиль написания программ конкретным программистом. Подобные преобразования делают уязвимыми методы, основанные на простых алгоритмах машинного обучения, работающих с множеством признаков, подконтрольных программисту. Тем не менее, современные глубокие модели, подразумевающие самостоятельное извлечение признаков из

векторизованного текста программы, могут быть более устойчивыми к подобным преобразованиям за счет фокусировки на более глубоких, неподконтрольных программисту или эксперту признаках.

1.7 Выводы по главе

1. Результаты анализа работ в области идентификации автора текстовой информации для целей кибербезопасности подтверждают сложность и неоднозначность решаемых задач. Для эффективного решения задач определения автора естественно- и искусственно-языкового текста, его пола и возраста, эмоционального состояния и идеологии необходим анализ творческой, повседневной и профессиональной деятельности и порождаемой при этом текстовой информации.

2. Методологическая база для комплексного и эффективного интеллектуального анализа текста для кибербезопасности отсутствует. Большинство применяемых исследователями подходов являются морально устаревшими. В то время, как в исследованиях фигурируют популярные алгоритмы SVM, АСД, RNN и CNN, число глубоких архитектур на базе трансформных моделей (BERT, GPT и др.) и число их гиперпараметров активно растут. На базе таких трансформерных моделей создаются атаки на методы, способные запутать классификатор, снизить точность и, более того, свести ее к нулю. Актуальные подходы должны быть устойчивы к таким атакам за счет своей способности к глубокому анализу неподконтрольных автору и неявных признаков, не поддающихся целенаправленному искажению. Процесс разработки методов определения авторства должен охватывать апробацию как зарекомендовавших себя state-of-the-art (SOTA) алгоритмов, так и современных глубоких НС, включая их двунаправленные вариации, комбинации архитектур и ансамбли.

3. Каждая задача, направленная на установление авторства, требует индивидуального подхода. Набор авторских признаков, на который опирается метод принятия решения, может иметь разную эффективность в зависимости от типа задачи,

анализируемых текстовых данных, типа языка (аналитический или синтетический). Методы, работающие с высокой эффективностью для англоязычных текстов, необходимо существенно модернизировать для других языков с учетом их особенностей. Различия в типах и жанрах текстов, объеме и числе доступных для анализа образцов накладывают дополнительные ограничения на возможность применения методов определения авторства.

4. Методы интеллектуального анализа текста вне зависимости от решаемой задачи кибербезопасности должны учитывать семантику и контекст. Семантические признаки являются наиболее информативными. Они отражают краткосрочные и долгосрочные зависимости между языковыми сущностями, приверженность к определенным синтаксическим конструкциям и речевым оборотам, авторские оценочные суждения и идеологию. Контекст может быть особенно полезен при решении задач, имеющих сопутствующий контент – например, при анализе комментариев к посту в тематическом сообществе.

5. Исходные коды программ требуют применения принципиально иных техник. В отличие от обычных текстов исходные коды создаются в соответствии со строгими синтаксическими правилами и парадигмами программирования. В них зачастую используются известные реализации алгоритмов и шаблоны.

6. Количество образцов авторского текста и их длина могут оказывать существенное влияние на эффективность алгоритмов. Важно учитывать особенности, возникающие при работе с небольшим числом образцов или с образцами, имеющими ограниченную длину, и выбирать соответствующие методы.

7. Множество признаков, используемых для обучения алгоритмов, должно быть информативным. Для получения такого множества следует использовать методы фильтрации, направленные на отсев запутывающих или мало информативных признаков. Без использования фильтрации уменьшается статистическая надежность работы алгоритма, возрастает время его работы, а также увеличивается вычислительная сложность обучения.

На основе анализа предметной области сформулируем основную **цель работы как создание методологии идентификации автора текстовой информации, включая естественно-языковые тексты и исходные коды программ, для решения задач кибербезопасности**.

2 МЕТОДОЛОГИЯ ИДЕНТИФИКАЦИИ АВТОРА ТЕКСТОВОЙ ИНФОРМАЦИИ ДЛЯ РЕШЕНИЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ

2.1 Этапы создания методологии

Под **методологией** здесь и далее будем понимать совокупность моделей, алгоритмов, методов и методик, предназначенную для достижения цели работы – идентификации автора текстовой информации, включая естественно-языковые тексты и исходные коды программ, для решения задач кибербезопасности.

Под **методикой** будем понимать систему, включающую определенные элементы методологии с указанием значений их параметров и последовательности применения этих элементов для решения практической задачи.

Разработка методологии включает 4 этапа (рисунок 2.1).

1. Аналитический этап. Анализ существующих проблем и решений в области анализа текстов для задач кибербезопасности, формулирование решаемой научной проблемы, определение цели и задач исследования.

2. Методический этап. Разработка методик, учитывающих специфику выделенных задач текстового анализа. Базовые методики включают комплекс наиболее релевантных и эффективных подходов и инструментов и корректируется в соответствии со спецификой текстовых данных в более сложных прикладных методиках.

3. Этап разработки. Программная реализация методик и инструментов, направленных на решение специфических задач текстового анализа, связанных с информационной безопасностью.

4. Этап верификации и анализа результатов. Подбор параметров для элементов методологии при решении важных прикладных народно-хозяйственных задач на репрезентативных датасетах.



Рисунок 2.1 – Этапы создания методологии

2.2 Авторская методология идентификации автора для решения задач кибербезопасности

На основе проведенного в разд. 1 анализа разработана комплексная методология идентификации автора текстовой информации [290] для решения задач кибербезопасности.

Представленная на рисунке 2.2 методология оперирует множеством взаимосвязанных моделей, методов и алгоритмов для анализа естественно- и искусственно-языковых текстов и включает:

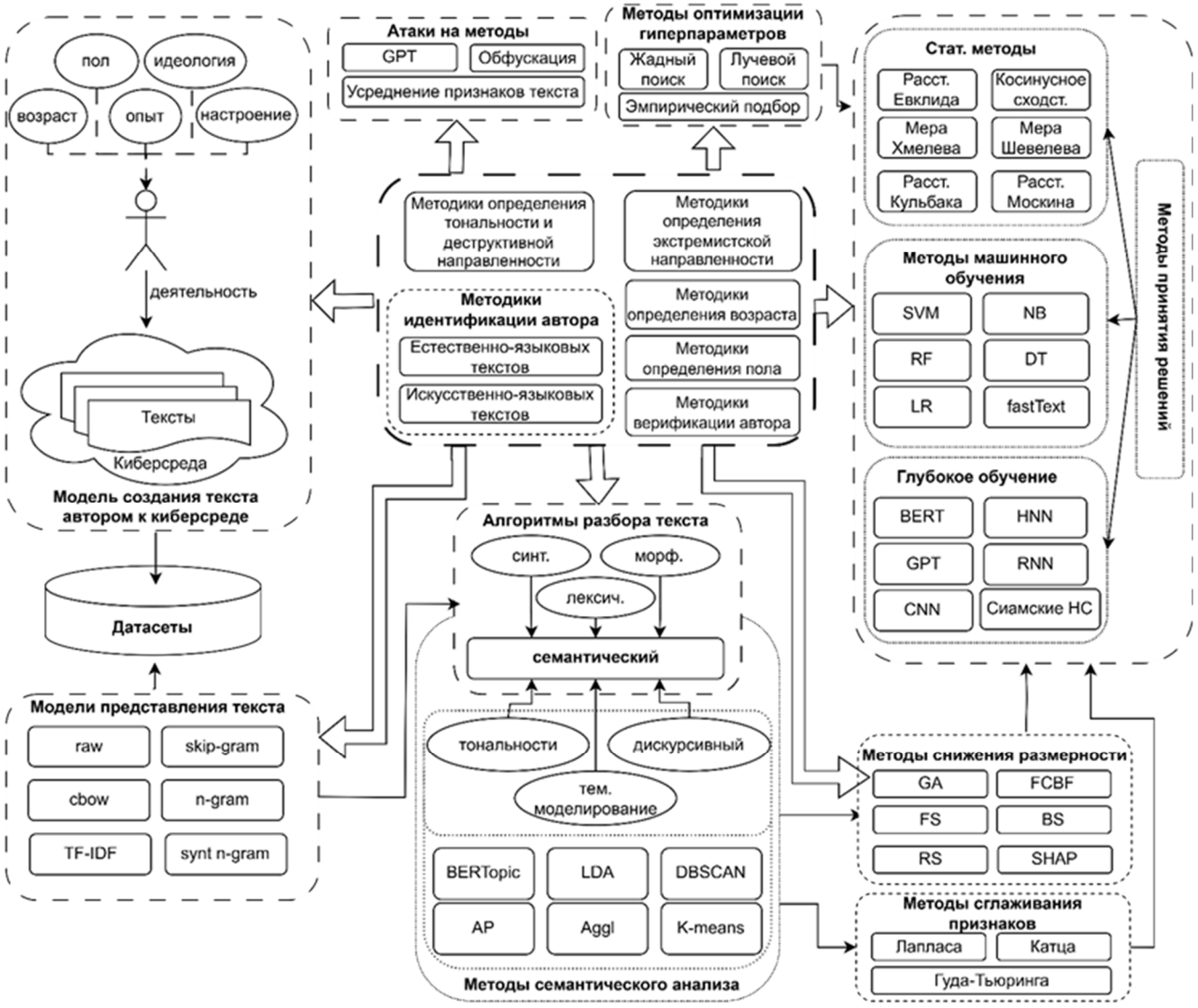


Рисунок 2.2 – Методология идентификации автора текстовой информации для решения задач кибербезопасности

1. Авторская модель создания автором текста в киберсреде. Ключевая модель методологии, описывающая процесс создания текстов авторами с учетом особенностей и ограничений среды. Описана в подразд. 2.3.

2. Модели представления текста. Тексты, подлежащие анализу, содержат множество явных и неявных признаков, указывающих на различные авторские характеристики: пол, возраст, опыт, идеологию, настроение и др. Каждый такой признак может оказать существенное влияние на конечный результат, поэтому важно представить текст в виде информативных признаков. Описаны в подразд. 2.4.

3. Алгоритмы разбора текста. Для разбора текста на различных уровнях (лексическом, морфологическом синтаксическом, семантическом) в рамках текстового анализа используются различные методы и инструменты машинного обучения и обработки естественного языка. Описаны в подразд. 2.5 и 2.6.

4. Методы принятия решений. Ключевой частью методологии является процесс принятия решения о принадлежности текста к классу в зависимости от задачи. Решение принимается статистическими методами, использующими меры расстояния и сходства признаков в пространстве, методами на основе машинного обучения, методами на основе глубокого обучения. Каждая группа методов имеет свои преимущества и недостатки по отношению к конкретной задаче текстового анализа. При этом методы могут применяться как по отдельности, так и совместно в качестве алгоритмического ансамбля. Описаны в подразд. 2.7.

5. Методы оптимизации гиперпараметров. Большинство задач машинного обучения сводятся к поиску параметров модели, которые минимизируют некоторую функцию потерь. Функция потерь оценивает, насколько хорошо модель соответствует данным. Благодаря оптимизации гиперпараметров, обучение становится более эффективным и быстрым, обеспечивается сходимость алгоритма и повышается качество конечной модели. Описаны в подразд. 2.8.

6. Методы снижения размерности и отбора признаков. Этот процесс помогает повысить эффективность и качество моделей машинного обучения. Во-первых, за счет снижения размерности уменьшается вычислительная сложность. Модели машинного обучения, особенно полученные путем глубокого обучения, могут быть

требовательны к вычислительным ресурсам. Уменьшение числа признаков может существенно сократить время обучения и предсказания, а также снизить требования к памяти. Во-вторых, применение информативных признаков в противовес полному множеству признаков позволяет улучшить обобщающую способность модели и предотвратить проблему переобучения за счет удаления нерелевантных или избыточных признаков. В-третьих, в результате снижения размерности признакового пространства повышается интерпретируемость модели: модели с меньшим числом признаков легче интерпретировать. Описаны в подразд. 2.9.

7. Атаки на методы. Атаки на методы – это специальные методы, предназначенные для запутывания моделей машинного обучения. Среди таких атак можно выделить обфускацию текста, искусственную-генерацию текста и усреднение признаков. Обфускация подразумевает добавление шума в виде опечаток, замену слов синонимами или бессмысленными словами, изменение порядка слов, применение гомоглифов (символов иноязычного алфавита, визуально похожих на исходный), а также использование парафразирования. Искусственная генерация позволяет выполнять имитацию стиля автора за счет использования современных генеративных моделей семейства GPT. С ее помощью создаются реалистичные тексты, способны запутать классификатор. Эти атаки могут быть статическими, созданными один раз и используемыми без изменений, или динамическими, адаптирующимися к обновлениям модели классификации. Применяемые для решения задач кибербезопасности модели классификации текстов должны быть устойчивыми к атакам на метод. Атаки на методы подробно описаны в подразд. 1.6.

8. Алгоритмы сглаживания. Информативные признаки текста, свойственные текстам больших объемов, могут не проявиться в текстах небольших объемов. Алгоритмы сглаживания позволяют оценить вероятности не наступивших событий. Описаны в подразд. 2.10.

9. Концептуальные различия естественно- и искусственно-языковых текстов представлены в подразд. 2.11.

2.3 Авторская модель создания автором текста в киберсреде с учетом семантики

Модель создания текста автором в киберсреде с учетом семантики представим тройкой:

$$\mathrm{M} = (\mathbf{A}, \mathbf{T}, \mathbf{E}), \tag{2.1}$$

где **A** – множество авторов; **T** – множество текстов; **E** – множество сред.

Пусть имеется множество текстов $\mathbf{T} = \{t_1, ..., t_{|\mathbf{T}|}\}$ и множество авторов $\mathbf{A} = \{a_1, ..., a_{|\mathbf{A}|}\}$. Введем бинарное отношение «текст написан автором» $R \subset \mathbf{T} \times \mathbf{A}$ на декартовом произведении множеств **T** и **A** такое, что выполняется *tRa* если текст $t \in \mathbf{T}$ написан автором $a \in \mathbf{A}$:

$$\exists t \in \mathbf{T}, \exists a \in \mathbf{A} : (tRa). \tag{2.2}$$

В случае, когда текст *t* можно представить как объединение фрагментов $t = \bigcup_{i=1}^{N_t} t'_i$, написанных несколькими авторами, будем говорить, что текст *t* «написан в соавторстве»:

$$\exists t'_i, t'_j \subseteq t, \exists a_l, a_m \in \mathbf{A}, a_l \neq a_m : \left(t'_i R a_l\right) \wedge \left(t'_j R a_m\right). \tag{2.3}$$

Автор может взаимодействовать или не взаимодействовать с другими людьми, оказывать на них влияние посредством написанных им текстов.

Случай, когда текст, написанный одним автором, подвергается изменению другим автором при сохранении общей семантики и тональности текста назовем «редактированием» и опишем в виде фнкции:

$$t^e = edit\left(t\right). \tag{2.4}$$

Текст, полученный в результате работы генеративного алгоритма, обученного на текстах $\mathbf{T}^a$ автора *a*, упрощенно опишем в виде:

$$t^g = gen\left(\mathbf{T}^a\right). \tag{2.5}$$

Обозначим факт вмешательства в процесс создания текста как

$$inter \in \mathbf{INTER} = \left\{ inter_1, ..., inter_{|\mathbf{INTER}|} \right\}, \quad (2.6)$$

где множеством активных действий по отношению к текстам является самостоятельное написание, редактирование, соавторство, применение генеративного алгоритма.

Каждый автор в момент создания текста представляется набором атрибутов, которые можно описать как:

$$\mathbf{C}^a = (id, age, gender, emo, status, action, pop) \in$$

$$\in \mathbf{ID} \times \mathbf{AGE} \times \mathbf{GENDER} \times \mathbf{EMOA} \times \mathbf{STATUSA} \times \mathbf{ACTION} \times \mathbf{POPA}, \quad (2.7)$$

где $id \in \mathbf{ID}$ – идентификатор: любая последовательность символов, которой человек себя идентифицирует. В качестве идентификатора может выступать ФИО, псевдоним в социальных сетях или на портале сети Интернет и др.; $gender \in \mathbf{GENDER} = \left\{ gender_1, ..., gende_{r|\mathbf{GENDER}|} \right\}$ – пол и гендерная идентичность: мужской, женский, представитель ЛГБТ*; $emo \in \mathbf{EMOA} = \left\{ emo_1, ..., emo_{|\mathbf{EMO}|} \right\}$ – настроение, с которым автор писал текст: положительное, отрицательное, нейтральное. Возможна более детальная классификация, учитывающая оттенки агрессии и др.; $status \in \mathbf{STATUSA} = \left\{ status_1, ..., status_{|\mathbf{STATUSA}|} \right\}$ – статус, показывающий имеет ли гражданин статус иностранного агента или экстремиста согласно [282], [280], или имеет отношение к организациям, имеющим эти статусы; $action \in \mathbf{ACTION} = \left\{ action_1, ..., action_{|\mathbf{ACTION}|} \right\}$ – деятельность автора по созданию текстовой информации. Можно выделить общение, творческую, научную и профессиональную деятельность; $pop \in \mathbf{POPA} = \left\{ pop_1, ..., pop_{|\mathbf{POPA}|} \right\}$ – категории популярности автора; $age \in \mathbf{AGE}$ – возраст автора.

Например, для решения практических задач кибербезопасности могут иметь значение возрастные интервалы:

* Запрещенная в Российской Федерации экстремистская организация.

– до 14 лет и старше, т.к. согласно п. 2 ст. 26 ГК РФ с 14 лет гражданин может заниматься любой интеллектуальной деятельностью в качестве правообладателя;

– до 16 лет и старше, так как 16 лет – возраст сексуального согласия согласно ч. 1 ст. 13 Семейного Кодекса РФ;

– до 18 лет и старше, так как с 18 лет у гражданина РФ наступает совершеннолетие согласно ст. 60 Конституции РФ;

– классификация Всемирной организации здравоохранения (ВОЗ): 18–44 лет – молодой возраст, 45–59 лет – средний возраст, 60–74 года – пожилой возраст, 75–90 лет – старческий возраст, старше 90 лет – долгожители;

– классификация согласно ст. 12 закона № 436-ФЗ от 29.12.2010 «О защите детей от информации, причиняющей вред их здоровью и развитию»: «0+», «6+», «12+», «16+», «18+» и др.

Популярность автора можно оценивать, например, по количеству подписчиков в социальных сетях; количеству проданных книг; индексу цитирования; количеству людей, у которых проект автора с исходными кодами находится в избранных проектах на хостинге IT-проектов и др. Авторов с высокой популярностью и рейтингом в киберсреде можно называть «лидерами мнений» [257].

Рассмотрим текст с точки зрения деления на классы, к которым его можно отнести, с учетом контекста кибербезопасности.

Текст имеет семантическое описание:

$$topic \in \mathbf{TOPICT} = \left\{ topic_1, ..., topic_{|\mathbf{TOPICT}|} \right\}. \quad (2.8)$$

Для портала Интернет примерами семантического описания могут быть «садоводство», «все о животных», «новости», «язык программирования C++», «научные статьи по защите информации» и др.

В свою очередь каждую тему можно представить списком ключевых слов (облаком тэгов):

$$topic_i = \left[keyword_1, ..., keyword_{|topic_i|} \right]. \quad (2.9)$$

Несомненно, с точки зрения лингвистики и филологии важными характеристиками является вид, стиль и жанр текста. Однако в контексте решения задач кибербезопасности в дальнейшем будем использовать абстрактное понятие «тип текста», которое учитывает вид, стиль, жанр, назначение текста и др.:

$$type \in \mathbf{TYPET} = \left\{ type_1, ..., type_{|\mathbf{TYPET}|} \right\}. \quad (2.10)$$

Например, в качестве типов в зависимости от задачи будем понимать художественные, любительские и сообщения из социальных сетей; естественно-языковые и искусственно-языковые и т.д.

Текст может быть отнесен к экстремистским материалам или материалам иностранных агентов, доступ к которым запрещен на территории Российской Федерации [271], [283]. Действующие статусы можно представить как:

$$status \in \mathbf{STATUST} = \left\{ status_1, ..., status_{|\mathbf{STATUST}|} \right\}. \quad (2.11)$$

Каждый текст имеет эмоциональный окрас (тональность):

$$emo \in \mathbf{EMO} = \left\{ emo_1, ..., emo_{|\mathbf{EMO}|} \right\}. \quad (2.12)$$

Эмоциональный окрас в простом случае может принимать значения: положительный, негативный, нейтральный. Возможна более детальная классификация, учитывающая оттенки радости, злости, грусти, страха, интереса и т.д.

Текст может получить популярность у целевой аудитории. Категории популярности можно представить как $pop \in \mathbf{POPT} = \{ pop_1, ..., pop_{|\mathbf{POPT}|} \}$.

Таким образом множество классов, к которым можно отнести текст, можно описать как декартово произведение вышеозначенных множеств, а конкретный класс, к которому относится текст, представить как:

$$\mathbf{C}^t = (type, status, topic, emo, inter, pop) \in$$
$$\in \mathbf{TYPET} \times \mathbf{STATUST} \times \mathbf{TOPICT} \times \mathbf{EMO} \times \mathbf{INTER} \times \mathbf{POPT}. \quad (2.13)$$

Далее рассмотрим текст с точки зрения составляющих его элементов.

Минимальной единицей текста является символ. Обозначим их множество как $\mathbf{L} = \left\{ l_1, ..., l_{|\mathbf{L}|} \right\}$. Последовательности символов, входящих в алфавит языка, образуют морфемы $\mathbf{Mo} = \left\{ m_1, ..., m_{|\mathbf{Mo}|} \right\}$, из которых в свою очередь состоят словоформы $\mathbf{W} = \left\{ w_1, ..., w_{|\mathbf{W}|} \right\}$. Несколько словоформ, находящихся в синтаксических связях, образуют словосочетания $\mathbf{C} = \left\{ c_1, ..., c_{|\mathbf{C}|} \right\}$. Словоформы и символы знаков препинания образуют предложения $\mathbf{S} = \left\{ s_1, ..., s_{|\mathbf{S}|} \right\}$. Одно или более предложений, объединенных по какому-то принципу, образуют параграфы $\mathbf{P} = \left\{ p_1, ..., p_{|\mathbf{P}|} \right\}$.

Тогда текст *t* можно представить в виде иерархического соподчинения последовательностей элементов:

$$t = \left\{ p_j^i \right\}_{i=1}^{N_p} = \left\{ s_j^i \right\}_{i=1}^{N_s} = \left\{ c_j^i \right\}_{i=1}^{N_c} = \left\{ w_j^i \right\}_{i=1}^{N_v} = \left\{ m_j^i \right\}_{i=1}^{N_m} = \left\{ l_j^i \right\}_{i=1}^{N_a}, \qquad (2.14)$$

где $l_j^i \in \mathbf{L}$; $m_j^i \in \mathbf{Mo}$; $w_j^i \in \mathbf{W}$; $c_j^i \in \mathbf{C}$; $s_j^i \in \mathbf{S}$; $p_j^i \in \mathbf{P}$; N_l; N_m; N_w; N_c; N_s; N_p – количество символов, морфем, слов, словосочетаний, предложений, абзацев в тексте.

Таким образом, текст можно рассматривать как иерархическую структуру [297] и анализировать на любом уровне как последовательность отдельных составляющих его элементов, групп элементов длиной *N* или их признаков [285] (рисунок 2.3).

Сложность анализа и автоматизации повышается с уровнем иерархии. На каждом из этапов могут возникнуть неточности. Например, из-за зашумленности анализируемого текста, омонимии, семантической неоднозначности и других особенностей русского языка. При статистическом анализе единиц текста действует правило, что чем выше уровень признака, тем больший объем текста нужен для стабилизации статистик.

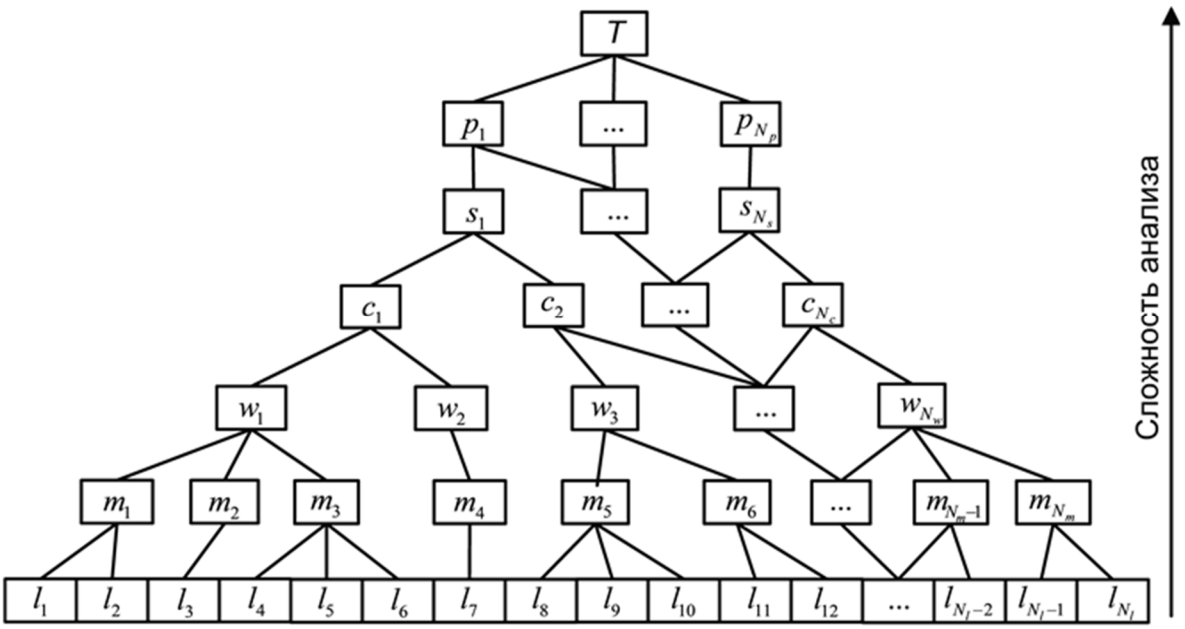


Рисунок 2.3 – Иерархическая структура текста

Каждый элемент текста описывается вектором признаков, отражающим его свойства. Для естественных текстов у слова, например, можно определить часть речи, морфологические признаки и длину и т.д. Для исходных кодов программ можно определить тип токена, семантику оператора и др.

Набор признаков текста можно представить как результат работы функции извлечения полного вектора характеристик из текста:

$$\mathbf{F} = extract(t) = \left[f_1, ..., f_{|\mathbf{F}|} \right], \tag{2.15}$$

где $f_i \in \mathbf{LEX} \cup \mathbf{MORPH} \cup \mathbf{SYNT} \cup \mathbf{SEM} \cup \mathbf{IDIO} \cup \mathbf{META} \cup \mathbf{EMB}$; $\mathbf{LEX} = \left[lex_1, ..., lex_{|\mathbf{LEX}|} \right]$ – лексические признаки; $\mathbf{MORPH} = \left[morph_1, ..., morph_{|\mathbf{MORPH}|} \right]$ – морфологические признаки; $\mathbf{SYNT} = \left[synt_1, ..., synt_{|\mathbf{SYNT}|} \right]$ – синтаксические признаки; $\mathbf{SEM} = \left[sem_1, ..., sem_{|\mathbf{SEM}|} \right]$ – семантические признаки; $\mathbf{IDIO} = \left[idio_1, ..., idio_{|\mathbf{IDIO}|} \right]$ – идиосинкразические признаки; $\mathbf{META} = \left[meta_1, ..., meta_{|\mathbf{META}|} \right]$ – метаданные текста; $\mathbf{EMB} = [emb_1, ..., emb_l]$ – некоторое векторное представление текста, полученное с помощью модели машинного обучения, где l – размер входного слоя модели.

Множество информативных характеристик текста $\mathbf{F}^{inf}$ и их значения зависят от типа текста и атрибутов автора, рассматриваемых в рамках интересующей задачи идентификации. Получение информативных признаков можно представить в виде функции, принимающей на вход текстовые признаки, тип текста и атрибуты автора:

$$\mathbf{F}^{inf} = inf(\mathbf{F}, type, \mathbf{C}^{a}) = \left[f_1^{inf}, ..., f_{\left|\mathbf{F}^{inf}\right|}^{inf} \right]. \quad (2.16)$$

Текстами и совокупностью векторов признаков текстов, написанных авторами, все или некоторые атрибуты которых совпадают, можно представить стиль определенной категории авторов.

$$\forall a_l, a_m \in \mathbf{A}, \mathbf{K}^{a_l} = \mathbf{K}^{a_m} = \mathbf{K}, \mathbf{K} \subseteq \mathbf{C}^{\mathbf{A}}:$$
$$\mathbf{C}^{\mathbf{K}} = \left\{ \mathbf{F}^{inf_i^{\mathbf{K}}} \right\}_{i=1}^{N_{\mathbf{T}^{\mathbf{K}}}} = \begin{Vmatrix} f^{inf_{1,1}^{\mathbf{K}}}, & \dots & f^{inf_{1,\left|\mathbf{F}^{inf\mathbf{K}}\right|}^{\mathbf{K}}} \\ \dots & \dots & \dots \\ f^{inf_{N_{\mathbf{T}^{\mathbf{K}}},1}^{\mathbf{K}}} & \dots & f^{inf_{N_{\mathbf{T}^{\mathbf{K}}},\left|\mathbf{F}^{inf\mathbf{K}}\right|}^{\mathbf{K}}} \end{Vmatrix}, \quad (2.17)$$

где $\mathbf{K}$ – подмножество совпадающих атрибутов авторов; $\mathbf{T}^{\mathbf{K}}$ – множество текстов, написанных авторами с одинаковыми атрибутами, а $N_{\mathbf{T}}{}^{\mathbf{K}}$ – количество текстов в этом множестве, $\mathbf{F}^{inf_i^{\mathbf{K}}}$ – множество значений лексических, морфологических, синтаксических, семантических, идиосинкразических признаков и эмбеддингов, информативных для определенной категории авторов, имеющих одинаковые атрибуты.

Например, матрица признаков для авторов возрастной категории «18+» мужского пола примет вид:

$$C^{18+,\text{муж}} = \left\{ \mathbf{F}^{inf_i^{\mathbf{18+,муж}}} \right\}_{i=1}^{N_{\mathbf{T}^{\mathbf{18+,муж}}}} = \begin{Vmatrix} f^{inf_{1,1}^{\mathbf{18+,муж}}} & \dots & f^{inf_{1,\left|\mathbf{F}\right|}^{18+,\text{муж}}} \\ \dots & \dots & \dots \\ f^{inf_{N_{\mathbf{T}^{\mathbf{18+,муж}}},1}^{\mathbf{18+,муж}}} & \dots & f^{inf_{N_{\mathbf{T}^{\mathbf{18+,муж}}},\left|\mathbf{F}\right|}^{\mathbf{18+,муж}}} \end{Vmatrix}. \quad (2.18)$$

Стиль автора может меняться со временем, и значения характеристик **F** могут изменяться. Однако множество информативных признаков $\mathbf{F}^{inf}$ должно быть устойчивым к этим изменениям во времени и учитывать небольшое редактирование другими авторами.

Среда, в которой автор пишет или публикует текст, описывается атрибутами:

$$\mathbf{C}^{e} = (topic, type, status, rules, pop) \in$$

$$\in \mathbf{TOPICE} \times \mathbf{TYPEE} \times \mathbf{STATUSE} \times \mathbf{RULES} \times \mathbf{POPE}, \qquad (2.19)$$

где $topic \in \mathbf{TOPICE} = \left\{ topic_1, ..., topic_{|\mathbf{TOPICE}|} \right\}$ – семантическое описание среды, т.е. список тематик, тексты, относящиеся к которым, размещаются в среде; $type \in \mathbf{TYPEE} = \left\{ type_1, ..., type_{|\mathbf{TYPEE}|} \right\}$ – тип среды. Например, Интернет-библиотека, мессенджер, социальная сеть, хостинг IT-проектов, Интернет-СМИ, сайт научного журнала, локальный компьютер автора и др.; $status \in \mathbf{STATUSE} = \left\{ status_1, ..., status_{|\mathbf{STATUSE}|} \right\}$ – действующий статус ресурса (разрешен или запрещен) в «Едином реестре доменных имен, указателей страниц сайтов в информационно-телекоммуникационной сети Интернет и сетевых адресов, позволяющих идентифицировать сайты в информационно-телекоммуникационной сети Интернет, содержащие информацию, распространение которой в Российской Федерации запрещено» [247]; $rule \in \mathbf{RULES} = \left\{ rule_1, ..., rule_{|\mathbf{RULES}|} \right\}$ – правила размещения текста в среде и/или стандарты среды и необходимо ли им следовать. Например, к ним можно отнести соблюдение законодательства РФ, правил публикации статей в научном журнале, стандартов кодирования для исходных текстов программ на хостинге IT-проектов в компании, запрет публикации текстов, не соответствующих определенной возрастной категории и др.; $pop \in \mathbf{POPE} = \left\{ pop_1, ..., pop_{|\mathbf{POPE}|} \right\}$ – популярность среды, каждый тип среды имеет свою специфику оценки. Например, для Интернет-ресурса – количество посещающих его людей в единицу времени; для группы в социальной сети или мессенджере – количество реальных подписчиков;

для сайта научного журнала – его импакт-фактор и т.д. Отметим, что чем популярнее среда, тем больший охват аудитории она имеет и тем быстрее опубликованная текстовая информация находит читателя.

Введем бинарное отношение «текст создается в среде» $Q \subset \mathbf{T} \times \mathbf{E}$ на декартовом произведении множеств **T** и **E** такое, что выполняется *tQe* если текст $t \in \mathbf{T}$ создается и размещается в среде $e \in \mathbf{E}$:

$$\exists t \in \mathbf{T} \exists e \in \mathbf{E} : (tQe). \tag{2.20}$$

Решение задач кибербезопасности (рисунок 2.4) сводится к отнесению текста *t* к классу $c \in \mathbf{C} = \mathbf{C}^t \bigcup \mathbf{C}^{\mathbf{K}}$ с учетом ограничений и особенностей среды $\mathbf{C}^e$ и авторских признаков $\mathbf{C}^a$ на основе текстов, класс для которых известен $\mathbf{T}' = \{t_1, ..., t_m\} \subseteq \mathbf{T}$, т.е. существует множество пар «текст-класс» $\mathbf{D} = \left\{\left(t_i, c_j\right)\right\}_{i=1}^{m}$. При этом каждый текст рассматривается как вектор признаков **F**. Целью является построение классификатора, решающего данную задачу, т.е. нахождение некоторой целевой функции $\Phi : \mathbf{T} \times \mathbf{C} \to [0,1]$. Значения функции интерпретируется как степень принадлежности объекта классу: 1 соответствует полностью положительному решению, 0 – отрицательному.



Рисунок 2.4 – IDEF0 диаграмма процесса анализа текста
для решения задач кибербезопасности

Примеры ограничений среды, накладываемых на авторов и создаваемые тексты, а также влияние атрибутов друг на друга попрано и в целом приведены в таблицах 2.1 и 2.2.

Таблица 2.1 – Примеры ограничения и влияния атрибутов попарно

Атрибут 1	Атрибут 2	Ограничения и влияния Атрибутов
t.topic	*e.topic*	Тематики текста и среды должны совпадать
t.type	*e.type*	В определенных типах сред (научные журналы, литературные блоги) отдается предпочтение определенным жанрам или форматам текстов
t.emo	*e.rules*	Эмоциональная окраска текста не должна противоречить правилам среды касательно тональности текста
t.status	*e.status*	Тексты, опубликованные на неразрешенных ресурсах, могут подвергаться цензуре или блокировке
t.status	*e.type*	Некоторые среды могут специализироваться на контенте, который в других средах может быть запрещен
a.age	*e.rules*	Среда может ограничить доступ к своему контенту на основе атрибута «возраст пользователя»
a.action	*e.rules*	Деятельность автора должна соответствовать правилам среды
a.status	*e.status*	Лица, признанные иноагентами или экстремистами, не должны осуществлять публикацию в разрешенных средах. Тексты таких авторов могут подвергаться дополнительной модерации
a.pop	*e.pop*	Популярные авторы могут привлечь аудиторию в среду и повысить ее популярность. Популярность автора может повыситься за счет популярности среды
a.gender	*t.topic*	Возраст и предпочтения автора могут влиять на выбор тематики текста
a.action	*t.type*	Профессиональная деятельность может определять тип и структуру текста
a.emo	*t.emo*	Настроение автора в момент написания может отражаться на эмоциональной окраске текста

Таблица 2.2 – Примеры ограничений и влияния атрибутов автора, текста и среды

Атрибут текста	Атрибут автора	Атрибут среды	Ограничения и влияние атрибутов
t.topic	*a.age*	*e.rules*	Возрастные ограничения среды ограничивают тематику текстов, которые может публиковать или читать автор определенного возраста
t.type	*a.gender*	*e.topic*	Некоторые среды более популярны среди определенного пола, что может влиять на выбор жанра или типа текста автором
t.emo	*a.action*	*e.type*	Профессиональная деятельность автора коррелирует с выбором профессиональных сред и специализированных тем, что делает тексты более строгими по тональности
t.status	*a.status*	*e.status*	Статус автора как иноагента и статус среды как разрешенного или запрещенного ресурса напрямую влияют на легальность и доступность текста для чтения
t.status	*a.action*	*e.rules*	Авторы, занимающиеся творческой деятельностью, могут сталкиваться с ограничениями в средах со строгими правилами относительно контента
t.topic	*a.pop*	*e.pop*	Популярные авторы выбирают тематики текстов, соответствующие интересам большинства пользователей популярной среды, чтобы увеличить свое влияние и расширить аудиторию
t.type	*a.action*	*e.type*	Авторы-ученые часто публикуют научные работы в специализированных журналах, где тип среды соответствует типу текста
t.status	*a.status*	*e.rules*	Авторы текстов, признанных иноагентами, могут сталкиваться с дополнительными ограничениями в определенных средах из-за строгих правил

Таким образом, была разработана модель создания автором текста в киберсреде, учитывающая атрибуты автора, текста и среды, важные при решении задач кибербезопаности.

2.4 Используемые модели представления текста

Представления текстов, также называемые эмбеддингами и встраиваниями, берут свое начало из идеи о том, как можно представить текстовую информацию в числовом виде без потери смысла. Обобщенная классификация моделей представления текста приведена на рисунок 2.5.

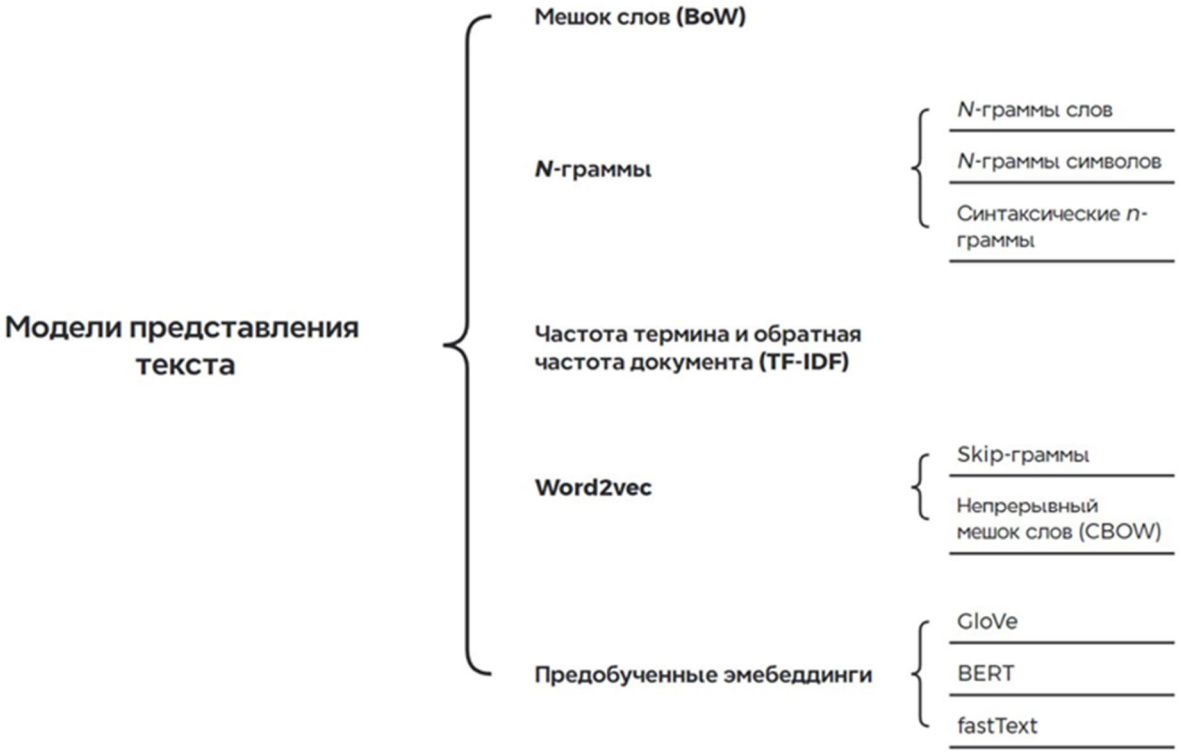


Рисунок 2.5 – Классификация моделей представления текстовой информации

2.4.1 Модель «мешка слов» (BOW)

Модель BOW [137] позволяет представить текст как изолированные слова, не принимая во внимание смысловые связи и порядок. Формально этот процесс можно описать следующим образом. Пусть **T** – это множество текстов, подлежащих анализу, а $\mathbf{V}_i$ – словарь i-го текста, который можно определить как $\mathbf{V}_i = \left\{ w_i^1, ..., w_i^{n_i} \right\}$, где n_i – это число слов, являющихся уникальными. В таком случае

словарь уникальных слов для всего множества текстов является объединением множеств $\{\mathbf{V}_i\}, i \in \mathbf{T}$:

$$\mathbf{W} = \bigcup_{i \in \mathbf{M}} \mathbf{V}_i = \{w \mid w \in \mathbf{V}_i, \exists i \in \mathbf{T}\}. \quad (2.21)$$

Размер вектора для *i*-го текста соответствует мощности словаря $|\mathbf{W}|$, а его элементы $\mathbf{H}_i = \left(h_1, ..., h_{|\mathbf{W}|}\right)$ формируются по следующему принципу:

$$h_k = \begin{cases} 1 \Leftrightarrow w_i^j \in \mathbf{W} \\ 0 \Leftrightarrow w_i^j \notin \mathbf{W} \end{cases}, j = \overline{1, n_i}, k = \overline{1, |\mathbf{W}|}\Bigg\}. \quad (2.22)$$

Модель BOW применялась во многих исследованиях [101, 160, 270, 272, 301] в рамках экспериментов с целью сравнения разных моделей представления текстовых данных для выбора наиболее эффективного в контексте решения конкретной задачи интеллектуального анализа текстов. Несмотря на популярность модели, она имеет проблемы (потеря контекста и порядка слов), которые делают ее менее эффективной в сравнении с другими. Для решения проблем были разработаны другие модели представления текстов, основанные на *n*-граммах, частотных и других статистических мерах, а также НС.

2.4.2 Модели на основе *n*-грамм

***N*-граммы** [57] представляют собой последовательно идущие символы, слова или их характеристики длиной *n*, анализируемые как отдельная единица текста (рисунок 2.6). Принцип работы статистического анализа *n*-грамм заключается в том, чтобы выявить наиболее часто встречающиеся или значимые последовательности слов или символов в тексте. Это позволяет понять структуру текста, выявить часто встречающиеся выражения или фразы, а также использовать эти знания для различных задач, таких как автоматическое извлечение ключевых слов, классификация текста, предсказание следующего слова и других.

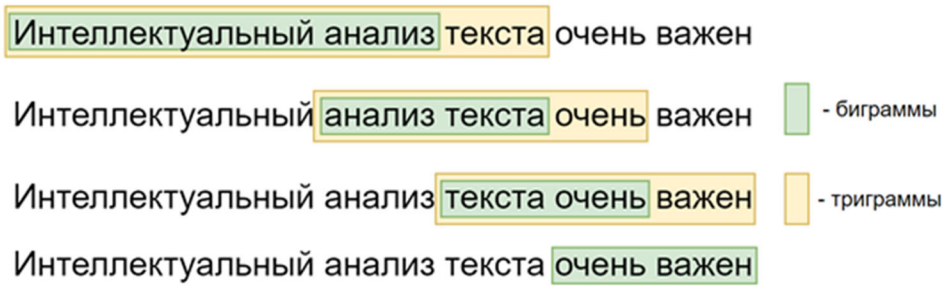


Рисунок 2.6 – Пример *n*-грамм

Математически *n*-граммы можно описать следующим образом. Пусть **A** – алфавит признаков, а $\mathbf{T} = \left(a_1,...,a_{L+(n-1)}\right)$ – составленная по нему последовательность, где L и n – положительные целые числа. В таком случае *n*-граммой будет цепочка последовательных признаков *n*. Тогда *i*-я *n*-грамма представляется как $\left(a_i,...,a_{i+n-1}\right)$, а последовательность **T** содержит L таких *n*-грамм.

В качестве алфавита **A** могут подразумеваться как символы, так и слова. С помощью последовательности символов можно моделировать механизмы образования слов, а с помощью *n*-грамм слов – деривацию предложений и текста в целом.

Если L – общее число подсчитанных *n*-грамм, а $C\left(a_i,...,a_{i+n-1}\right)$ – число появлений *n*-граммы $\left(a_i,...,a_{i+n-1}\right)$, то при достаточно больших L частоты:

$$f\left(a_i,...,a_{i+n-1}\right) = \frac{C\left(a_i,...,a_{i+n-1}\right)}{L}. \tag{2.23}$$

Относительная частота $f\left(a_i,...,a_{i+n-1}\right)$ в отдельных случаях считается приближением вероятности $P\left(a_i,...,a_{i+n-1}\right)$ появления конкретной *n*-граммы в случайной позиции текста.

Точнее процессы, происходящие в языке, позволяет описать модель, предполагающая, что появление конкретного признака зависит от предшествующих ему признаков:

$$P\left(a_1,...,a_L\right) = P\left(a_1\right)P\left(a_2 \mid a_1\right)\prod_{i=3}^{L} P\left(a_i \mid a_1,...,a_{i-1}\right). \tag{2.24}$$

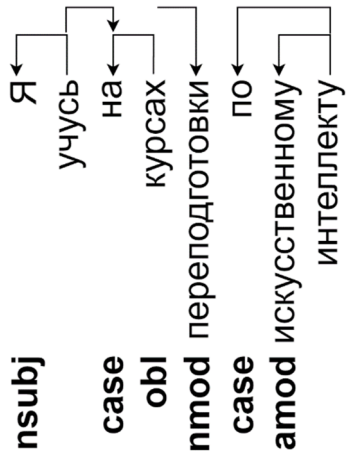


Рисунок 2.7 – Принцип работы синтаксических *n*-грамм

Помимо стандартных *n*-грамм, для представления текстовых данных используются **синтаксические *n*-граммы** (рисунок 2.7). Они учитывают синтаксическую структуру слов в предложении, что позволяет брать во внимание связи между словами, их роль в предложении и синтаксическую грамматику и получать синтаксическое дерево в результате анализа.

Модели на основе простых *n*-грамм и синтаксических *n*-грамм популярны у исследователей ввиду их высокой эффективности и простоты реализации [26, 29, 38, 58, 83, 113, 135, 160, 164, 167, 216, 239, 256].

Стоит отметить, что *n*-граммные модели позволяют описывать текстовые структуры, имеющие фиксированную длину.

2.4.3 Модель на основе TF-IDF

В то время, как применение *n*-грамм позволяет сохранить контекст и порядок, отдельные задачи требуют оценки важности отдельно взятых слов. В таких случаях эффективной моделью является TF-IDF [203], представленной впервые в 1986 г.

TF-IDF [142, 155, 181, 195, 203, 270, 272, 305], представляет собой статистическую метрику, отображающую важность слова в контексте документа или коллекции документов. В отличие от BoW и *n*-грамм, метрика TF-IDF позволяет учитывать частоту слова и понижать значимость широко употребляемых слов в контексте конкретного анализа. Формально метрику можно описать как:

$$TFIDF = TF \cdot IDF = \frac{n_t}{\sum\limits_{i=1}^{k} n_i} \cdot \ln\left(\frac{n_c}{\sum\limits_{j=1}^{m} n_j}\right), \tag{2.25}$$

где n_t – количество отдельно взятых слов в документе или коллекции; $\sum_{i=1}^{k} n_i$ – общее число слов в документе или коллекции; $\sum_{j=1}^{m} n_j$ – общее число документов или коллекций, в которых наблюдается искомое слово.

2.4.4 Модели на основе Word2Vec

В условиях необходимости сохранения семантического значения текстовой информации будут эффективны модели Word2Vec [38, 142, 165, 233, 244, 272, 274, 301], к которым относятся непрерывный BOW (CBOW) и skip-граммы.

Модели BOW и CBOW, несмотря на схожие названия, являются концептуально разными методами. Модель BOW представляет текст как набор слов и их частот в данном документе или корпусе. Она не учитывает порядок или контекст, в котором появляются слова, и может не передать полного смысла текста. Модель BOW проста и легка в реализации, но ограничена в возможностях передачи смысла. Модель **CBOW**, впервые описанная Т. Миколовым в 2013 г. [67], напротив, представляет собой подход на основе НС, который фиксирует контекст слов. Она учится предсказывать целевое слово на основе слов, которые появляются до и после него в данном контекстном окне. Учитывая окружающие слова, модель CBOW может лучше улавливать значение слова в данном контексте.

Применение модели CBOW может быть описано следующими шагами:

1. В качестве входных данных CBOW берет на вход контекст целевого слова.

2. Преобразование в векторы: каждое слово в окне анализа слева и справа представляется one-hot вектором, которые затем конкатенируются во входной вектор. Под one-hot понимается представление категориальных данных в виде бинарных векторов. В случае работы с набором слов, каждое слово представляется в виде вектора, в котором все элементы равны нулю, за исключением одного элемента, соответствующего индексу этого слова в словаре, который равен единице. Пример работы метода представлен на рисунке 2.8.

Символ	One-hot вектор
i	[105] 0,0,0,…,0,0,1,0,…,0,0,0,0
n	[110] 0,0,0,…,1,0,0,0,…,0,0,0,0
t	[116] 0,0,0,…,0,1,0,0,…,0,0,0,0
[space]	[32] 0,0,0,…,0,0,0,1,…,0,0,0,0
m	[109] 0,0,0,…,0,0,1,0,…,0,0,0,0
a	[97] 0,0,0,…,1,0,0,0,…,0,0,0,0
i	[105] 0,0,0,…,0,1,0,0,…,0,0,0,0
n	[110] 0,0,0,…,0,0,0,1,…,0,0,0,0

Рисунок 2.8 – Принцип работы метода one-hot encoding

3. Обучение НС: CBOW использует НС с одним скрытым слоем, чтобы предсказать целевое слово на основе контекста.

4. Получение векторного представления: после обучения НС выдает веса скрытого слоя, которые и являются векторным представлением слова.

Skip-граммы, также представленные в [67], в противовес CBOW, использующему контекст для предсказания слова, используют целевое слово для предсказания окружающих слов. Шаги работы метода схожи с CBOW, отлична лишь работа однослойной НС. Математически skip-граммы можно описать следующим образом. Для получения матрицы скрытого слоя необходимо умножить one-hot вектор центрального слова x на первую весовую матрицу **W**:

$$\mathbf{H} = \mathbf{W}^T x\,. \tag{2.26}$$

Для получения матрицы выходного слоя **U** выполняется умножение скрытого слоя **H** на вторую весовую матрицу **W’**:

$$\mathbf{U} = \mathbf{W}'^T \mathbf{H}\,. \tag{2.27}$$

Затем нужно применить функцию активации SoftMax к слою **U**, чтобы получить выходной слой y. Пусть u_j – j-й нейрон слоя u, w_j – j-е слово в словаре, где j – любой индекс, $\mathbf{V}_{wj}$ – j-й столбец матрицы $\mathbf{W}'$, тогда:

$$u_j = \mathbf{W}'^T_{w_{i,j}} \mathbf{H}\,, \tag{2.28}$$

$$y_j = softmax\left(u_j\right). \tag{2.29}$$

Здесь y_j обозначает вероятность того, что w_j является контекстным словом:

$$P\left(w_i \mid w_j\right) = y_j = \frac{e^{u_j}}{\sum\limits_{j'=1}^{v} e^{u_{j'}}}, \tag{2.30}$$

где $P\left(w_j \middle| w_i\right)$ – вероятность того, что w_j – контекстное слово, учитывая, что w_i – входное слово. Цель состоит в максимизации $P\left(w_{j^*} \middle| w_i\right)$, где j^* представляет индексы контекстных слов:

$$\prod_{c=1}^{C} \frac{e^{u_{j^* c}}}{\sum\limits_{j'=1}^{v} e^{u_j}}, \tag{2.31}$$

где j^*_c – словарные индексы контекстных слов. Контекстные слова варьируются как $c = \overline{1, C}$, где C – число контекстных слов.

Полученные skip-граммы и CBOW векторы представляют собой плотные векторы фиксированной длины, которые содержат семантическое описание слова на основе его контекста в корпусе текста. Векторы со схожим значением или семантической информацией располагаются в пространстве близко друг к другу.

2.4.5 Модели на основе предобученных эмбеддингов

Представления текстов в виде предобученных эмбеддингов являются неотъемлемой составляющей современных подходов и методик к решению задач интеллектуального анализа текста [24, 26, 38, 143, 150, 160, 225, 230, 241, 272].

Предобученные эмбеддинги – представления слов, которые были извлечены из больших корпусов и доступны для повторного использования в различных задачах текстового анализа. Такие представления фиксируют семантические отношения между словами, позволяя модели осмысленно понимать сходства и отношения между разными словами. Можно выделить несколько наиболее популярных из них: GloVe, fastText, BERT.

GloVe, презентованная Д. Пеннингтоном в 2014 г. [169], обучена работе с глобальной статистикой совпадения слов. Она использует глобальный контекст для создания представлений слов, которые отражают общее значение слов на основе вероятностей их совместного появления. Создание матрицы совпадения слов является фундаментальным компонентом GloVe. Эта матрица обеспечивает количественную меру семантической близости между словами, фиксируя частоту, с которой они появляются вместе в данном контексте. Затем, минимизируя разницу между скалярным произведением векторов и точечной взаимной информацией соответствующих слов, GloVe оптимизирует векторы слов. GloVe способна создавать плотные векторные представления, фиксирующие синтаксические и семантические отношения:

$$J = \sum_{i,j}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2, \tag{2.32}$$

где V – размер словаря; X_{ij} – входная последовательность; w – вектор слова; $\tilde{w}$ – вектор контекстных слов; b – смещение для w, $\tilde{b}$ – смещение для $\tilde{w}$.

Принцип работы эмбеддингов **fastText**, описанный в 2016 г. [74], основан на идее использования CBOW и разбиения слов на подслова. Алгоритм fastText начинает с построения словаря из всех уникальных слов и подслов в обучающем корпусе. Затем каждое слово представляется в виде суммы эмбеддингов его подслов. Далее происходит обучение однослойной НС с использованием метода градиентного спуска для минимизации функции потерь. В результате обучения формируются векторные представления слов, которые учитывают контекст и подслова. Основное преимущество fastText заключается в том, что он способен работать с редкими словами, а также учитывать морфологические особенности языка. Кроме того, fastText позволяет строить эмбеддинги для слов, которых не было в обучающем корпусе, путем комбинации эмбеддингов их подслов. Подробнее о математическом аппарате fastText в подразд. 2.7.

Принцип работы эмбеддингов **BERT**, разработанных Я. Девлингом и др. в 2018 г. [30], основан на использовании трансформеров, которые способны улавливать контекстуальные зависимости между словами в предложении. Как и fastText,

BERT обучается на огромном объеме текстовых данных, путем предсказания отсутствующих слов в предложениях. Процесс обучения BERT включает в себя два основных этапа: предобучение и дообучение. Во время предобучения модель обучается на больших неразмеченных корпусах текстов, а затем дообучается на конкретной задаче (классификация текста или извлечение информации). BERT использует многоуровневую архитектуру трансформера, которая позволяет модели улавливать контекстуальные зависимости между словами в предложении благодаря механизму самообучения модели. Преимуществом BERT является способность учитывать контекст и семантические связи слов в тексте. Подробная информация о математическом аппарате BERT представлена в подразд. 2.7.

2.5 Используемые алгоритмы разбора текста

Разбор текста осуществляется на лексическом, синтаксическом, морфологическом и семантическом уровнях (рисунок 2.9).

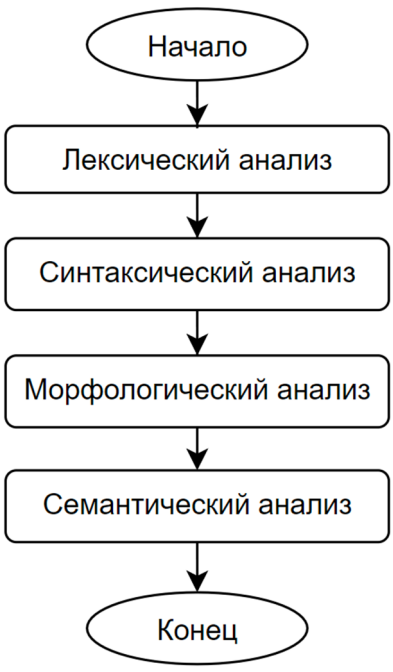


Рисунок 2.9 – Этапы анализа текста

На лексическом уровне анализируется значение слов и выражений вне контекста. Такой анализ может выполняться путем выявления часто встречающихся сочетаний слов (анализ коллокаций) или путем изучения значения слов и их отношения друг к другу (лексическая семантика). Лексический анализ осуществляется путем разбиения текста на токены (символы, слова, фразы), приведения токенов к корневой форме за счет удаления окончаний и использования словарей для точного поиска начальной формы слова.

На морфологическом уровне происходит определение характеристик слов, таких как часть речи, падеж, число, время и др. Если слово найдено в словаре, то для него возвращаются все доступные характеристики, если нет, то выполняется предсказание на базе имеющихся знаний, полученных из обучающих данных и статистических данных о суффиксах и окончаниях. Полученные предсказания ранжируются согласно их вероятностям.

Синтаксический анализ обеспечивает понимание структуры текста, помогает в анализе способа, которым слова в предложении сочетаются друг с другом, формируя фразы и предложения. На этапе разбора предложения строится его синтаксическое дерево, где узлы представляют слова или фразы, а ребра – отношения между ними. На этапе применения грамматических правил анализируется структура предложений на основе таких подходов как контекстно-свободные грамматики или зависимые грамматики.

Для выполнения лексического, синтаксического и морфологического анализа существует множество инструментов на базе различных алгоритмов. Наиболее эффективны для русского языка: Spacy [202], Stanford NLP [204], NLTK [159] и др.

Семантический уровень фокусируется на понимании значения слов, фраз и предложений, а также на том, как эти элементы сочетаются для передачи смысла в более широком контексте. Наибольший интерес для решения задач кибербезопасности представляет тематическое моделирование, подробно представленное в подразд. 2.6. Оно направлено на обнаружение скрытых тематических структур в больших объемах текстовых данных и позволяет выявлять ключевые тематики, рассматриваемые в тексте.

2.6 Используемые методы семантического анализа

Методики и инструменты на основе алгоритмов кластеризации, в частности, семантической, могут применяться для тематического моделирования [101, 157, 233, 239, 254]. Критически значимой проблемой, требующей применения тематического моделирования на основе семантической кластеризации, является выявление негативно окрашенной информации и токсичной речи в текстах, передаваемых в сети Интернет. Крайней степенью такой эмоциональной окраски текстов являются запрещенные законодательством РФ пропаганда экстремизма и терроризма, призывы к розжигу межнациональной, религиозной и другой ненависти и розни. Идентификация таких текстов является актуальной задачей кибербезопасности.

Данная проблема имеет множество методов решения, которые можно разделить на группы по принципу действия (рисунок 2.10).

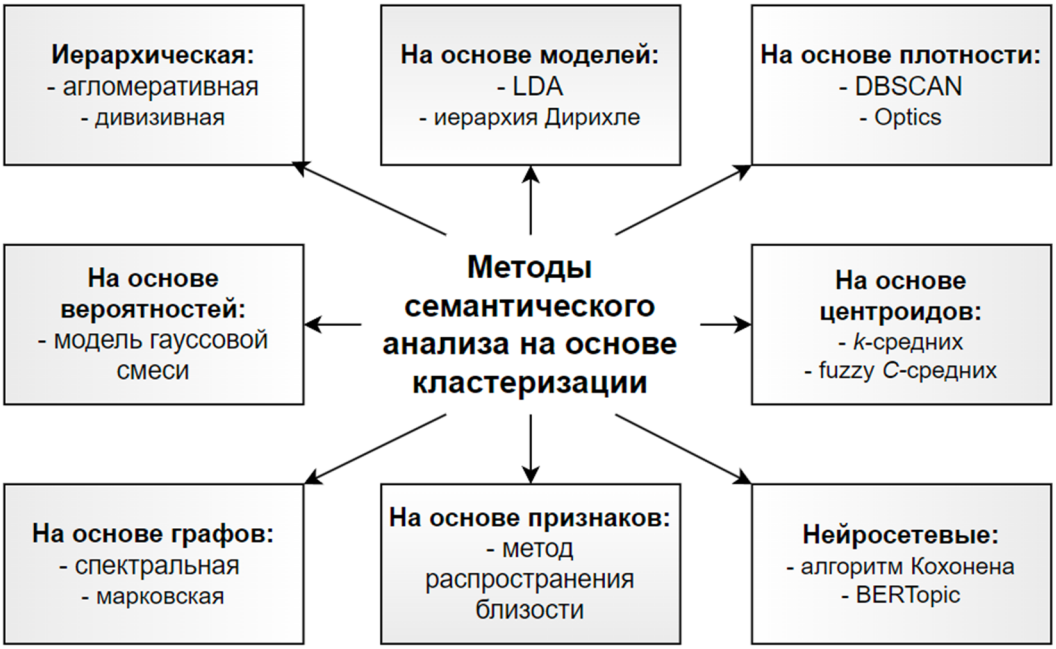


Рисунок 2.10 – Классификация методов кластеризации текста

2.6.1 Методы иерархической кластеризации

Методы иерархической кластеризации [267] строят иерархию, где каждый кластер может содержать подкластеры. Примером такого метода является агломеративная кластеризация. **Алгомеративная кластеризация** объединяет близкие объекты и постепенно объединяет более удаленные объекты, пока все объекты не будут объединены в один кластер или пока не будет достигнуто определенное условие останова. Предположим, есть два кластера *r* и *s*, и эти два кластера объединены в новый кластер *t*. Пусть *k* – любой другой существующий кластер, тогда расстояние между новым кластером *t* и существующим кластером *k:*

$$D(t,k) = \min\big(D(r,k), D(s,k)\big), \tag{2.33}$$

где $D(t,k)$ – минимальное расстояние между объектами в кластере *t* и *k*.

2.6.2 Методы на основе плотности объектов

Методы на основе плотности объектов анализируют расположение объектов в пространстве и определяют плотность вокруг каждого объекта. Объекты, находящиеся в областях более высокой плотности, считаются частью одного кластера, в то время как объекты в областях более низкой плотности считаются аномалиями. Эти методы могут быть особенно полезны для обнаружения кластеров различных форм и размеров. Например, пространственная кластеризация приложений с шумом на основе плотности (**DBSCAN**). Принцип работы DBSCAN основан на 3 идеях. Во-первых, это определение основных точек. Точка считается основной, если в ее окрестности содержится как минимум заданное количество точек, которые находятся на расстоянии не больше определенного значения. Во-вторых, определение граничных точек. Точка считается граничной, если она не является основной, но находится в окрестности основной точки. В-третьих, определение шумовых точек. Точка считается шумовой, если она не является ни основной, ни граничной.

2.6.3 Методы на основе центроидов

Методы на основе центроидов определяют кластеры, исходя из расстояния до центроидов. Таким методом является популярный алгоритм *k*-средних. Принцип работы кластеризации ***k*-средних** [136] заключается в разделении набора данных на заранее определенное количество кластеров (*k*) путем минимизации суммарного квадратичного отклонения точек от их центров кластеров. Для поиска центроида из кластерной группы точек применяется формула:

$$c_i = \frac{1}{|S|} \sum_{x_i \in S_i} x_i \,, \tag{2.34}$$

где S_i – набор всех точек, отнесенных к *i*-му кластеру; x_i – точка *i*-го кластера.

2.6.4 Методы на основе плотности вероятности

Методы на основе плотности вероятности осуществляют моделирование распределения вероятностей и поиск областей с наибольшей плотностью. Эти методы могут быть полезны, когда данные имеют вероятностную интерпретацию или когда требуется учет неопределенности в данных. Примером такого метода является **модель гауссовой смеси**. Процесс обучения включает определение параметров гауссовой смеси, таких как центроиды и матрицы ковариации, оценку вероятностей принадлежности к кластерам, присвоение текстов к кластерам, обновление параметров гауссовой смеси на основе новых присвоений. Обучение длится до момента, пока не сойдется к оптимальным параметрам или до достижения заданного критерия останова. Формально процесс обучения для многомерного случая можно описать как:

$$p\left(\vec{x}\right) = \sum_{i=1}^{K} \varphi_i N\left(\vec{x} \,|\, \vec{\mu}_i, \sum i\right), \tag{2.35}$$

$$N\left(\vec{x}\,|\,\vec{\mu}_i,\sum i\right)=\frac{1}{\sqrt{(2\pi)^K\left|\sum i\right|}}\exp\left(-\frac{1}{2}\left(\vec{x}-\vec{\mu}_i\right)^T\sum i^{-1}\left(\vec{x}-\vec{\mu}_i\right)\right), \tag{2.36}$$

$$\sum_{i=1}^{K}\varphi_i=1, \tag{2.37}$$

где K – число компонентов; μ_i – математической ожидание i; φ – вес компонентов смеси, x_i – точка данных.

2.6.5 Методы на основе графов

Методы на основе графов представляют данные в виде графа, где объекты являются вершинами, а связи между объектами – ребрами. Кластеры определяются на основе связей в графе. Один из таких методов – **спектральная кластеризация**, использующая спектральные свойства матрицы схожести для выделения кластеров. Алгоритм спектральной кластеризации можно описать следующим образом:

1. Исходя из данных, строится матрица схожести, которая отражает степень похожести между объектами.

2. На основе матрицы схожести строится матрица Кирхгофа – взвешенный граф, где веса ребер соответствуют степени схожести между объектами.

3. Для матрицы Кирхгофа вычисляются собственные векторы и соответствующие им собственные значения.

4. На основе собственных векторов и значений выполняется алгоритм кластеризации, например, метод *k*-средних.

5. Выделение кластеров на основе значений собственных векторов.

Алгоритм Ши – Малика [148] является одной из наиболее популярных техник спектральной кластеризации. Он разбивает точки на два множества $\left(B_1,B_2\right)$ исходя из собственного вектора v, соответствующего второму по величине собственному значению симметрично нормализованной матрицы Кирхгофа, вычисляемой как:

$$L^{norm}=I-D^{-1/2}AD^{-1/2}, \tag{2.38}$$

$$D_{ij} = \sum_{j} A_{ij} , \tag{2.39}$$

где A_{ij} – матрица схожести; I – матрица идентичности.

Спектральная кластеризация обладает рядом преимуществ, таких как способность обрабатывать сложные нелинейные структуры данных, а также возможность выделения кластеров произвольной формы.

2.6.6 Методы на основе плотности признаков

Методы на основе плотности признаков учитывают не только расстояние между объектами, но и плотность распределения признаков, описывающих эти объекты. Эти методы могут быть полезны, когда значения признаков более информативны, чем пространство признаков, например, в случае временных рядов или текстовых данных. Пример – **метод распространения близости (AP)**. Данный метод основан на идее распространения «ответственности» и «доступности» между точками данных. Шаги алгоритма метода распространения близости:

1. Каждая пара точек данных обменивается сообщениями об «ответственности» и «доступности». «Ответственность» отражает, насколько точка считает другую точку центром кластера, а «доступность» отражает насколько точка готова принять другую точку в качестве центра кластера.

2. Сообщения обновляются на основе текущих значений «ответственности» и «доступности» для каждой пары точек данных.

3. На основе значений выбираются центры кластеров.

4. Каждая точка данных присваивается к ближайшему центру кластера на основе значений «ответственности».

5. После присвоения точек кластерам сообщения об «ответственности» и «доступности» обновляются с учетом новых присвоений.

Эти шаги повторяются до тех пор, пока процесс не сойдется к оптимальным центрам кластеров или до достижения заданного критерия останова.

Преимущество метода заключается в заранее заданном числе кластеров и их автоматическом определении на основе входных данных.

2.6.7 Методы на основе статистических моделей

Методы на основе статистических моделей моделируют данные с использованием статистических моделей и определяют кластеры, основываясь на параметрах моделей. Примером метода является **Латентное размещение Дирихле** (LDA) [165]. LDA не предполагает какой-либо меры расстояния между кластерами. Вместо расстояния LDA использует подсчет слов методом BoW. Математически работу метода можно описать:

$$P\left(z_i = j \mid \mathbf{z}_{-i}, \mathbf{w}\right) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \cdot \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}, \tag{2.40}$$

где $P\left(z_i = j \mid \mathbf{z}_{-i}, \mathbf{w}\right)$ отражает вероятность назначения темы j i-му слову, что зависит от вероятности слова w_i в теме j и вероятности темы j в документе d_i; $n_{-i,j}^{(w_i)}$ – количество присвоений w_i теме j; $n_{-i,j}^{(\cdot)}$ – количество слов в теме j; $n_{-i,j}^{(d_i)}$ – количество появлений темы j в документе d_i; $n_{-i,\cdot}^{(d_i)}$ – количество появлений тем в d_i; T – число топиков; W – число слов в словаре; α, β – гиперпараметры Дирихле.

2.6.8 Семантические нейросетевые методы

Семантические нейросетевые методы используют НС для кластеризации. Нейросетевые методы могут быть основаны на самоорганизующихся картах, ассоциативных НС или глубоком обучении. Они позволяют автоматически извлекать сложные закономерности и структуры из данных, что может быть особенно полезно для больших и сложных наборов данных. Пример такого метода – **BERTopic**.

BERTopic (рисунок 2.11) основан на использовании предобученной модели BERT для извлечения семантических векторных представлений текстовых документов.

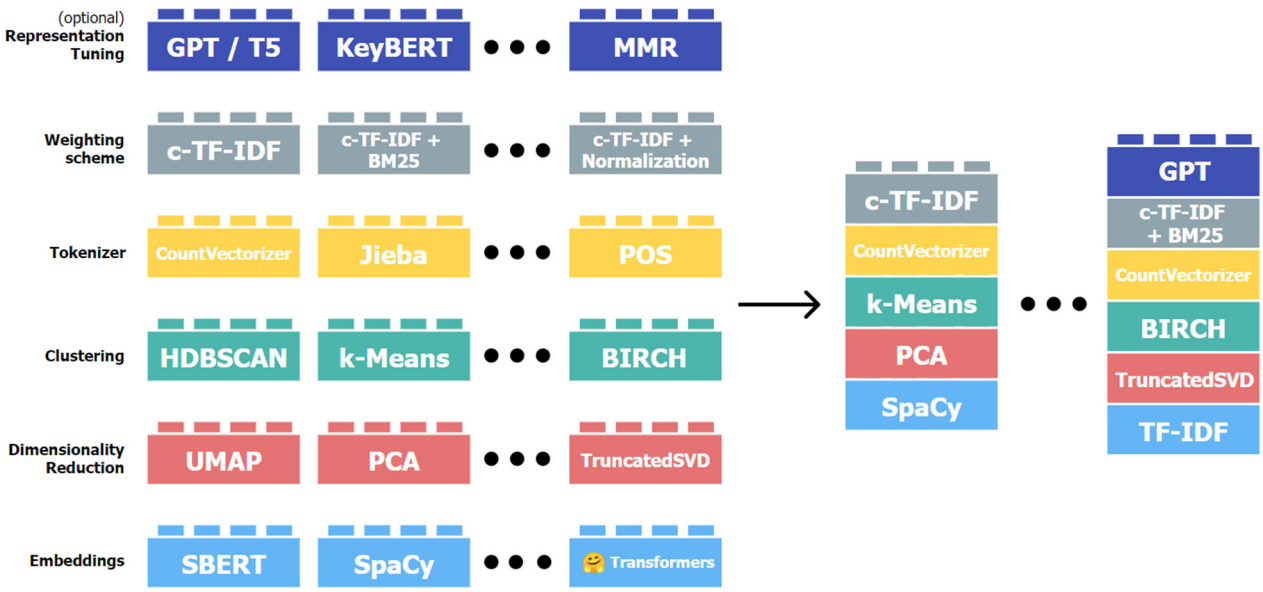


Рисунок 2.11 – Устройство BERTopic

Работа метода BERTopic осуществляется в несколько этапов:

1. Каждый текстовый документ преобразуется в векторное представление с использованием предобученной модели BERT (опционально, может быть использована другая глубокая модель). Это позволяет получить представления текстов, отражающие семантические связи и учитывающие контекст.

2. На основе векторных представлений документов строится граф схожести, где вершины представляют документы, а ребра отражают степень схожести между ними. Обычно используются метрики сходства, такие как косинусное сходство.

3. На основе построенного графа схожести применяется алгоритм кластеризации для выделения кластеров. Алгоритм кластеризации выбирается в соответствии с требованиями конкретной задачи.

4. Выделение тематических кластеров, которые отражают семантическую структуру коллекции документов.

Эффективность описанных в подразд. 2.6 методов напрямую зависит от области применения, специфики данных, а также условий применения этих методов.

2.7 Используемые методы принятия решений

Раздел посвящен обзору наиболее эффективных методов интеллектуального анализа текста, их математическим аппаратам, принципам работы, преимуществам и недостаткам. Рассмотрению подлежат методы, представленные в классификации по принципу действия (рисунок 2.12), в частности, статистические, на основе НС и классического машинного обучения, а также на основе глубокого обучения.

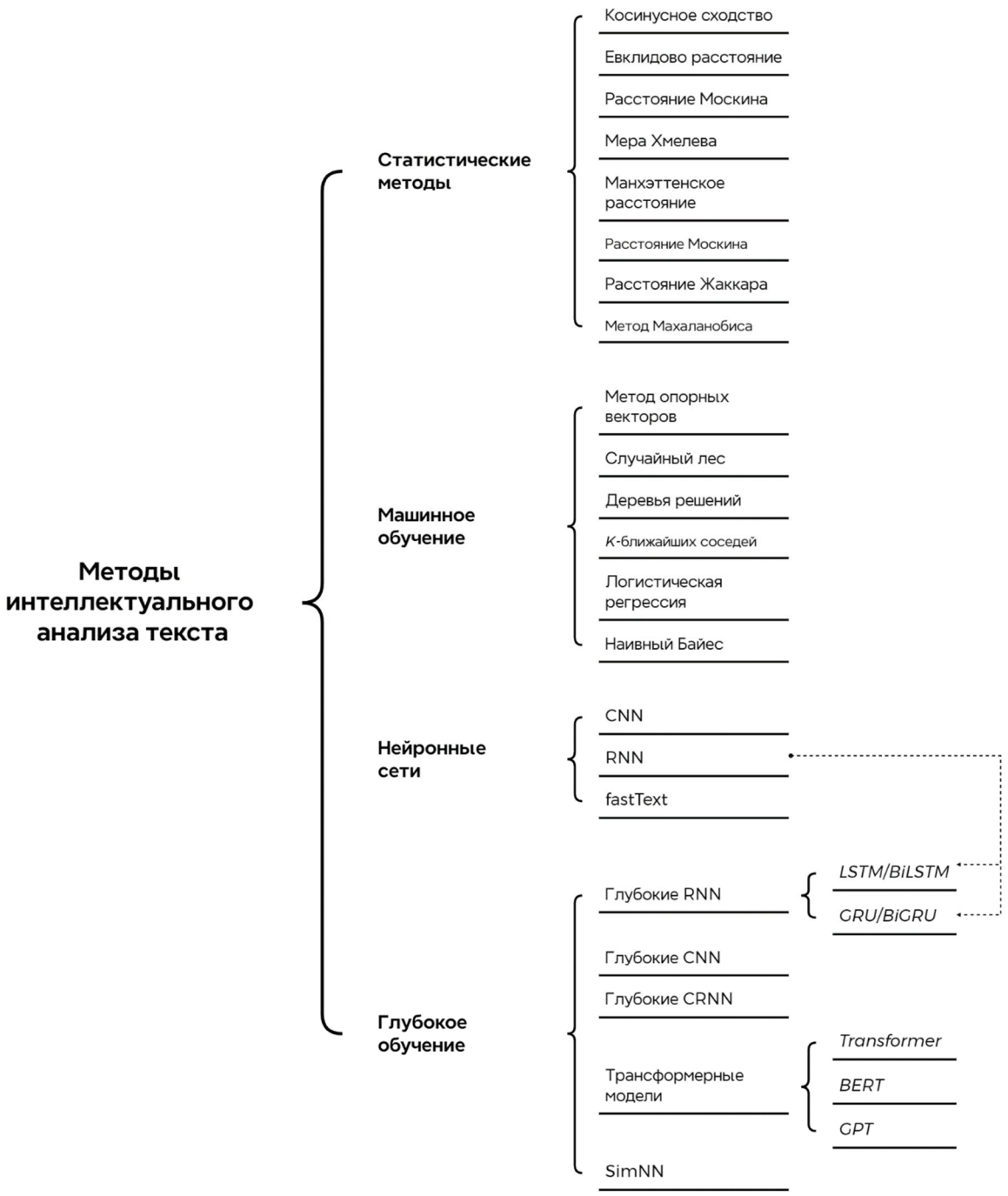


Рисунок 2.12 – Методы интеллектуального анализа текста

2.7.1 Методы на основе статистики

Методы на основе статистики применяются на протяжении десятков лет, начиная с середины XX века. Они являются наиболее простыми для понимания, реализации и интерпретации и являются базой для методов текстового анализа, включая методы машинного обучения и нейросетевые [110, 128, 164, 252, 274].

В контексте анализа текста можно выделить несколько наиболее известных и распространенных: расстояния Кульбака, Евклида, косинусное сходство и др.

Метод на основе расстояния Евклида измеряет фактическое расстояние между двумя векторами **X** и **Y** в многомерном пространстве:

$$d_{euc}(\mathbf{X},\mathbf{Y}) = \sqrt{\sum_{i=1}^{n}\left(x_i - y_i\right)^2}\,, \tag{2.41}$$

где $x_i \in \mathbf{X}$, $y_i \in \mathbf{Y}$.

Метод косинусного сходства базируется на скалярном произведении векторов и представляется как:

$$d_{\cos}(\mathbf{X},\mathbf{Y}) = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \cdot \sqrt{\sum_{i=1}^{n} y_i^2}}\,. \tag{2.42}$$

Косинусное сходство варьируется от –1 до 1, где –1 указывает на полную противоложность, 1 – полное совпадение, а 0 – ортоганальность.

Еще одним методом на основе расстояния является **метод Махаланобиса** [146]. Расстояние Махаланобиса – мера расстояния между двумя точками в многомерном пространстве, учитывающая корреляцию и ковариацию между признаками. Формула расстояния между двумя случайными точками *U* и *V*, одна из которых может принадлежать некоторому классу *C* с матрицей ковариаций *COV*:

$$d_{Mah}\left(U,V,COV^{-1}\right) = \sqrt{\left(U-V\right)\cdot COV^{-1}\cdot\left(U-V\right)^T}\,, \tag{2.43}$$

$$cov_{a,b} = \frac{1}{|C|-1}\sum_{X\in C}\left(X_a - \mu_a\right)\cdot\left(X_b - \mu_b\right), \tag{2.44}$$

где μ_a и μ_b – математические ожидания *a* и *b* признаков, |C| – количество точек в классе **C**, а $cov_{a,b} \in COV$.

В методе на основе **манхэттенского расстояния** близость двух текстов, представленных векторами, можно рассчитать, как:

$$d_{man} = \sum_{i=1}^{n} \left| x_{T_i} - x_{C_i} \right|. \tag{2.45}$$

Среди таких методов можно выделить **расстояние на основе максимального общего подграфа**, разработанное Н.Д. Москиным. Предложенная им метрика была разработана в связи с тем, что рассмотренные в рамках диссертационной работы [274] меры расстояний не учитывали нумерацию вершин графов, играющую важную роль при сравнении теоретикографовых моделей текстов. Сначала Н.Д. Москиным рассматривается модификация метрик для сравнения графов с одинаковым числом вершин, когда $|G| = |F| = n$, тогда расстояние между графами *G* и *F*:

$$d_1(G,F) = 1 - \min_{i=1,\ldots,n} \left(\frac{\left| msc\left(g_i, f_i\right) \right|}{i} \right). \tag{2.46}$$

При этом при задании меры на множестве графов желательно, чтобы $d\left(G_i, G_j\right)$, удовлетворяла следующим свойствам метрики (для произвольных *i*, *j*, *k*): неотрицательность, симметричность, тождественность и равенство треугольников. Н.Д. Москин в своем исследовании [274] выдвигает две теоремы. Первая из них заключается в том, что расстояние d_1 удовлетворяет всем свойствам метрики и расстояние между графами *G* и *F* с разным числом вершин можно найти следующим образом (пусть для определенности $|G| = m, |F| = k, m < k$):

$$d_2(G,F) = 1 - \min_{i=1,\ldots,k} \left(\frac{\left| msc\left(g_{\min(i,m)}, f_i\right) \right|}{i} \right). \tag{2.47}$$

Вторая теорема заключается в том, что расстояние d_2 также удовлетворяет всем свойствам метрики. Тогда в общем случае расстояние на основе максимального общего подграфа рассчитывается как:

$$d_{Mos}(G,F) = 1 - \min_{i=1,\ldots,l}\left(\frac{\left|msc\left(g_{\min(i,m)}, f_i\right)\right|}{i}\right). \quad (2.48)$$

Статистические меры как методы решения задач классификации текстов исследовались Д.В. Хмелевым. Идея состоит в подсчете и обработке парных сочетаний элементов текста, составляющих две матрицы: текстовую и эталонную. **Мера Хмелева** [331] работает на основе матриц частот переходов (букв, слов или предложений и их свойств). В качестве меры сравнения двумерных распределений представляющих матрицы частот переходов она использует:

$$L = \sum_{i=1}^{k}\sum_{j=1}^{k} m_{1ij} \cdot \ln\left(\frac{m_{1ij}}{n_{1i}} \Big/ \frac{m_{2ij}}{n_{2i}}\right), \quad (2.49)$$

где m_{1ij} – число переходов из i элемента в j в тексте; n_{1i} – общее число переходов из i-го элемента; m_{2ij}, n_{2i} – аналогичные числа для матрицы того автора, с которым производится сравнение; k – число элементов (например, сочетаний букв). Чем меньше различие между матрицами, тем меньше значение L по модулю.

В работе [334] О.Г. Шевелевым предложены схожие меры, дающие лучшие результаты за счет усреднения по анализируемому тексту:

$$I = 2 \cdot \sum_{i=1}^{k}\sum_{j=1}^{k} \frac{m_{1ij}}{n_1} \cdot \ln\left(\frac{m_{1ij}}{n_{1i}} \Big/ \frac{m_{2ij}}{n_{2i}}\right), \quad (2.50)$$

$$I = 2 \cdot \sum_{i=1}^{k}\sum_{j=1}^{k} \frac{m_{1ij}}{n_1} \cdot \ln\left(\frac{m_{1ij}}{\sum_{i=1}^{k} n_{1i}} \Big/ \frac{m_{2ij}}{\sum_{i=1}^{k} n_{2i}}\right). \quad (2.51)$$

2.7.2 Методы на основе классических алгоритмов машинного обучения

Методы машинного обучения можно разделить на контролируемые (с учителем, неконтролируемые (без учителя), полуконтролируемые (с частичным привлечением учителя), а также методы с подкреплением. Однако данная диссертационная работа фокусируется на методах контролируемых.

Сначала модель анализирует известный набор обучающих данных и соответствующие экземплярам его классов метки. Процесс обучения состоит в перестройке внутренних параметров модели на основе обучающих данных таким образом, чтобы оптимизировать прогностические возможности – повысить метрики эффективности модели или снизить размер потерь.

Классические методы многообразны, однако наиболее популярными для решения задач интеллектуального анализа текста являются NB, SVM, DT, RF, LR [29, 119, 156, 164, 171, 222, 240, 267, 274, 318, 330, 333].

В основе **метода NB** лежит теорема Байеса. Цель состоит в предсказании вероятности принадлежности объекта классу на основе его признаков. Истинным классом, к которому принадлежит вектор признаков **X** длиной *n*, считается класс *y* с наибольшей вероятностью, полученной наивным Байесовским классификатором.

$$P(y \mid \mathbf{X}) = \frac{P(x_1 \mid y) P(x_2 \mid y) ... P(x_3 \mid y)}{P(x_1) P(x_2) ... P(x_n)}, \tag{2.52}$$

$$P(y \mid \mathbf{X}) = \frac{P(y) \prod_{i=1}^{n} (x_i \mid y)}{P(x_1) P(x_2) ... P(x_n)}, \tag{2.53}$$

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i \mid y). \tag{2.54}$$

Следующий метод классического машинного обучения особенно популярен для решения задач определения авторства текста. Математический аппарат **SVM** был предложен В.Н. Вапником [210]. SVM используется для создания бинарного линейного классификатора, работающего в пространстве признаков высокой размерности. Основная цель этого метода – максимизировать зазор между

ближайшими точками двух классов, которые называются опорными векторами. Многоклассовую классификацию можно свести к решению нескольких бинарных задач с использованием стратегий типа «один против всех», «каждый против каждого» или других подходов, что позволяет успешно классифицировать по нескольким классам при высокой точности решения бинарных задач.

Для того чтобы преобразовать исходные данные в пространство, где разделяющая поверхность становится линейной, применяется ядровая функция:

$$\left(\Phi(x),\Phi\left(x'\right)\right)=k\left(x,x'\right), \tag{2.55}$$

где $\Phi(x),\Phi\left(x'\right)$ – скалярное произведение между вектором тестируемого образца и векторами из обучающего множества; k – функция, отображающая исходные данные в пространство с возможностью линейного разделения.

Функция классификации в этом случае задается как:

$$f(x)=\left\{\sum_{i=1}^{\ell}\alpha_i y_i k\left(x_i,x\right)\right\}+b, \tag{2.56}$$

где α – оптимальный коэффициент; y – метка класса; b – параметр смещения.

Оптимальное значение коэффициента α находится путем максимизации целевой функции:

$$W(\alpha)=\sum_{i=1}^{\ell}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{\ell}y_i y_j\alpha_i\alpha_j k\left(x_i,x_j\right), \tag{2.57}$$

в положительном квадранте $0\le\alpha_i\le C$, $i=\overline{1,\ell}$. Условие максимизации:

$$\sum_{i=1}^{\ell}\alpha_i y_i=0. \tag{2.58}$$

Параметр регуляризации C, являющийся для данной задачи штрафной функцией, задает соотношение между размером зазора и количеством ошибок на обучающей выборке.

Для решения задачи нелинейного программирования необходимо выполнение условий Каруша – Куна – Такера (ККТ) для всех i:

$$\begin{aligned} &\alpha_i = 0 \Rightarrow y_i f\left(x_i\right) \geq 1, \\ &0 < \alpha_i < C \Rightarrow y_i f\left(x_i\right) = 1, \\ &\alpha_i = C \Rightarrow y_i f\left(x_i\right) \leq 1. \end{aligned} \tag{2.59}$$

Параметр *b* выбирается так, чтобы обеспечивать выполнение второго условия ККТ для всех максимизируемых множителей Лагранжа. Множители, которые после максимизации имеют ненулевые значения, формируют опорные вектора.

Линейное ядро, представленное выше, используется, когда данные линейно разделимы гиперплоскостью. В зависимости от задачи применяются также полиномиальное, гауссовское и сигмоидальное ядра для нелинейного случая.

Принцип работы **дерева решений** [153] состоит в создании древовидной структуры, каждый внутренний узел которой инициирует проверку признака, каждая ветвь представляет результат проверки, а каждый листовой узел (конечный узел) содержит метку класса. Среди наиболее известных алгоритмов можно выделить ID3, C4.5, дерево классификации и регрессии (CART), а также автоматический детектор взаимодействия хи-квадрат (CHAID).

Дерево создается путем рекурсивного разделения обучающих данных на подмножества на основе значений признаков до тех пор, пока не будет выполнен критерий останова (максимальная глубина дерева или минимальное количество выборок, необходимое для разделения узла).

В процессе обучения дерево решений выбирает лучший признак для разделения данных на основе определенного показателя. Цель состоит в том, чтобы найти признак, который максимизирует получение информации после разделения. В качестве показателя для разделения данных чаще всего используется энтропия. В случае классификации она измеряет случайность на основе распределения меток классов в наборе данных. Энтропия для подмножества исходного набора данных, имеющего количество классов *K* для *i*-го узла, может быть определена как:

$$H = -\sum_{k \in K}^{n} p(i,k) \log_2 p(i,k), \tag{2.60}$$

$$p(k) = \frac{1}{n} \sum I(y = k), \tag{2.61}$$

где S – образец набора данных; k – конкретный класс из K классов; $p(k)$ – это доля объектов данных, принадлежащих классу k, от общего количества объектов данных в выборке набора данных S.

Мера прироста информации определяет уменьшение энтропии или дисперсии, возникающее в результате разделения набора данных на основе определенного свойства. Она используется в алгоритмах дерева решений для определения полезности функции путем разделения набора данных на более однородные подмножества по меткам классов или целевой переменной. Чем выше прирост информации, тем более ценной является функция для прогнозирования целевой переменной. Прирост информации от признака A по отношению к набору данных S рассчитывается следующим образом:

$$InfoGain(H, A) = H - \sum \frac{|H_V|}{|H|} H_v, \tag{2.62}$$

где A – конкретный признак или метка класса; |H| – энтропия выборки набора данных S; $|H_v|$ – количество экземпляров в подмножестве S, имеющих значение v для атрибута A.

RF [39], представляет собой модель, состоящую из нескольких деревьев решений. Каждое дерево RF строится независимо от других на основе подмножества случайно выбранных признаков и/или случайно выбранных образцов данных. Такой подход помогает уменьшить корреляцию между деревьями, что делает RF более устойчивым к переобучению. Каждое дерево решений в RF выдает свой прогноз, а затем результаты агрегируются путем голосования для получения окончательного прогноза RF:

$$\hat{C}_{rf}^{B} = majority\_vote\left\{\hat{C}_b(x)\right\}_1^B, \tag{2.63}$$

где $\hat{C}_b(x)$ – предсказание отдельного дерева решений.

LR, предложенная Д. Коксом в 1973 г. [55] – это алгоритм, который преобразует выходные данные непрерывного значения функции линейной регрессии в выходные данные категориальных значений с помощью сигмовидной функции

(логистической функции), которая отображает любой набор входных независимых переменных с действительным знаком в значение от 0 до 1.

Еще один популярный метод машинного обучения – ***K*-ближайших соседей**, представленный в 1951 г. [80]. Алгоритм работает путем поиска *K* ближайших соседей к заданной точке данных на основе метрики расстояния, такой как Евклидово расстояние или Манхэттена. Конкретный метод выбора ближайших соседей может различаться, но общий подход заключается в сортировке расстояний в порядке возрастания и выборе *K* точек данных с наименьшими расстояниями. Метка точки данных определяется большинством голосов или средним значением *K* соседей. Значение *K* выбирается на основе входных данных. Если они содержат больше выбросов или шума, лучше использовать более высокое значение *K*.

2.7.3 Методы на основе нейронных сетей

Основой для развития НС послужила работа McCulloch и Pitts [145].

Среди современных методов машинного обучения, которые базируются на простых НС, можно отметить метод **fastText** [74]. Он способен эффективно работать с большими объемами текстовых данных, создавать компактные векторные представления слов и проводить обучение как с учителем, так и без него. Работа метода fastText осуществляется в несколько этапов:

1. Предварительная обработка текста: текст разделяется на отдельные слова или *n*-граммы символов, которые будут использоваться для создания векторных представлений. Эти представления усредняются таким образом, чтобы на весь текст приходилось лишь одно представление.

2. Обучение модели: НС с одним скрытым слоем, количество параметров которой равно произведению размера словаря на размер представления, обучается предсказывать целевые слова или метки классов на основе контекста. В процессе обучения модель создает векторные представления слов, учитывая их контекстуальные и морфологические особенности.

3. Использование векторных представлений: полученные векторные представления слов могут быть применены для различных задач обработки естественного языка.

Рассчитать степень принадлежности текста к определенному классу согласно принципу работы fastText можно по формуле:

$$\lg = -\frac{1}{N}\sum_{n=1}^{N} y_n \log\left(f\left(B\mathbf{A}x_n\right)\right), \tag{2.64}$$

где x_n – это представление слова в n-грамме; **A** – матрица, извлекающая представления слов; B – линейное преобразование выходных данных; f – функция SoftMax.

С появлением больших данных и увеличением вычислительной мощности стало возможным обучение более сложных моделей, таких как CNN и RNN.

RNN [188] – это НС, в которой выходные данные предыдущего шага подаются в качестве входных данных на текущем шаге. Это помогает запоминать предыдущее слово, когда нужно предсказать следующее слово в предложении для сохранения контекста.

Фундаментальным процессором в RNN является рекуррентный блок. Он обладает уникальной способностью сохранять скрытое состояние, позволяя сети фиксировать последовательные зависимости, запоминая предыдущие входные данные во время обработки.

Принцип архитектуры RNN отображен на рисунке 2.13, где h – скрытое состояние; **X** – входной вектор; U – функция активации; L – функция потерь; W – весовой коэффициент; V – информация предыдущего шага.

RNN состоит из нескольких фиксированных функциональных блоков активации – по одному на каждый временной шаг. Каждый модуль имеет скрытое состояние, хранящее прошлые знания. Оно обновляется на каждом временном шаге, чтобы обозначить изменение знаний сети о прошлом, с использованием следующего рекуррентного отношения:

$$h_t = f\left(h_{t-1}, x_t\right), \tag{2.65}$$

где h_t – текущее; h_{t-1} – предшествующее; x_t – входное состояния.

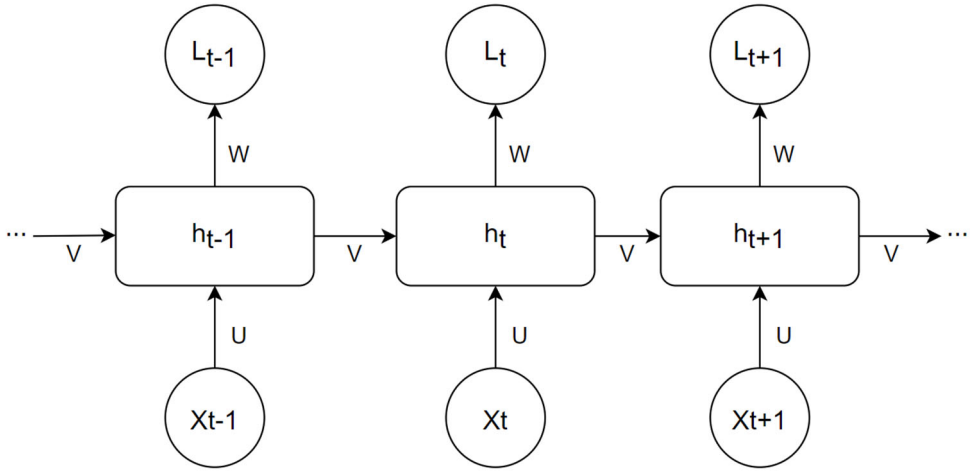


Рисунок 2.13 – Принцип работы RNN

С функцией активации формула примет вид:

$$h_t = \tanh\left(\mathbf{W}_{hh} h_{t-1} + \mathbf{W}_{hx} x_t\right), \tag{2.66}$$

где $\mathbf{W}_{hh}$ – матрица весов рекуррентного блока; $\mathbf{W}_{hx}$ – весовая матрица входной информации.

Тогда для выхода рекуррентного блока будет применяться формула:

$$y_t = \mathbf{W}_{hy} h_t, \tag{2.67}$$

где $\mathbf{W}_{hy}$ – матрица весов на выходном слое.

В процессе вычисления все параметры обновляются с помощью обратного распространения ошибки во времени (BBTT).

RNN легла в основу многих современных глубоких архитектур, таких как сеть LSTM, управляемый рекуррентный блок (GRU) и их двунаправленные вариации (BiLSTM, BiGRU).

Идея применения **CNN** [130], изначально предложенных для обработки изображений, к текстам заключается в использовании сверточных слоев для извлечения признаков из последовательности слов или символов с последующим применением полносвязных слоев для классификации или предсказания.

Конструкция CNN представляет собой НС прямого распространения, созданную путем сборки скрытых слоев друг над другом в определенном порядке. Последовательная конструкция позволяет CNN изучать иерархические атрибуты.

При этом размер исходных матриц CNN уменьшается за счет краевых эффектов. Под этим понятием подразумевается обработка информации на границах входных объектов. Когда CNN анализирует объект, краевые эффекты позволяют сети выделять границы и другие важные детали. В CNN краевые эффекты обнаруживаются с помощью операции свертки, извлекающей различные признаки из данных.

CNN с учетом краевых эффектов можно описать как:

$$x_j^l = f\left(\sum_i x_i^{l-1} * k_j^{\ l} + b_j^{\ l}\right), \tag{2.68}$$

где x_j^l – карта признаков j; $b_j^{\ l}$ – коэффициент сдвига для слоя l для карты признаков j; $k_j^{\ l}$ – ядро свертки входа x с ядром k.

Простые методы на основе НС являются оптимальным вариантом для решения большинства задач интеллектуального анализа текста [38, 68, 228, 241, 244]. Они сочетают в себе простоту реализации и интерпретации по аналогии с методами классического машинного обучения и преимущества нейросетевых архитектур, заключающиеся в возможностях извлечения информативных признаков разного масштаба и различных временных зависимостей. Тем не менее, для задач, где понимание контекста играет ключевую роль, более эффективными могут быть методы на основе глубоких нейронных архитектур.

2.7.4 Методы на основе глубокого обучения

Развитие глубокого обучения началось в конце 2000-х. Оно стало возможным благодаря появлению вычислительных мощностей, достаточных для обработки больших объемов данных. Кроме того, в тот же временной период была решена главная проблема классических НС, связанная с затухающими или взрывающимися градиентами, которая оказывала негативное влияние на эффективность ранних моделей. Проблема была решена в результате усовершенствования существующих рекуррентных и сверточных архитектур и появления таких методов, как

LSTM и GRU, пользующихся популярностью для решения задач интеллектуального анализа текста [9, 12, 24, 45, 143, 216, 230, 241, 305].

Глубокая **LSTM** [102] представляет собой модификацию RNN, способную эффективно обрабатывать долгосрочные зависимости в последовательности данных. LSTM решает проблему неспособности RNN к хранению долгосрочных связей. Кроме того, LSTM спроектирована таким образом, что проблема исчезающего/взрывающегося градиента практически полностью устранена, а модель обучения остается неизменной.

Формально состояния ячейки LSTM в момент времени *t* описываются как:

$$f_t = \sigma\left(\mathbf{X}_t * \mathbf{U}_f + H_{t-1} * \mathbf{W}_f\right), \tag{2.69}$$

$$\overline{C_t} = \tanh\left(\mathbf{X}_t * \mathbf{U}_c + H_{t\text{-}1} * \mathbf{W}_c\right), \tag{2.70}$$

$$I_t = \sigma\left(\mathbf{X}_t * \mathbf{U}_i + H_{t-1} * \mathbf{W}_i\right), \tag{2.71}$$

$$O_t = \sigma\left(\mathbf{X}_t * \mathbf{U}_o + H_{t-1} * \mathbf{W}_o\right), \tag{2.72}$$

где f_t – состояние гейта забывания; $\mathbf{X}_t$ – входной вектор; H_{t-1} – скрытое состояние предыдущей ячейки; C_{t-1} – состояние памяти предыдущей ячейки; H_t – скрытое состояние текущей ячейки; C_t – состояние памяти текущей ячейки; $\mathbf{W}_f$, $\mathbf{U}_f$ – векторы весов для гейта забывания; $\mathbf{W}_c$, $\mathbf{U}_c$ – векторы весов для гейта кандидатов; $\mathbf{W}_i$, $\mathbf{U}_i$ – векторы весов для входного гейта; $\mathbf{W}_o$, $\mathbf{U}_o$ – векторы весов для выходного гейта; σ – сигмоидальная функция.

Ключевым элементом, отличающим LSTM от RNN, у которых только 1 тангенсоидный слой, является состояние памяти $\overline{C_t}$, состоящее из 3 логистических сигмовидных гейтов и 1 тангенсоиднго слоя. Оно определяет необходимость запоминания или забывания информации и существенно влияет на значение гейта забывания, который принимает значения в диапазоне [0;1], где 0 означает полное забывание предыдущей информации, а 1 – запоминание.

На основе текущего состояния памяти $\overline{C_t}$ можно вычислить следующее:

$$C_t = f_t * C_{t-1} + I_t * \overline{C}_t, \tag{2.73}$$

где C_t – текущее состояние памяти на шаге *t*.

Выходное значение ячейки LSTM формально можно описать как:

$$H_t = O_t * \tanh(C_t). \tag{2.74}$$

Полученные состояния передаются на следующий временной шаг и процесс повторяется.

Принцип работы **GRU** [73] во многом схож с LSTM. Отличительной особенностью GRU является закрытый механизм для эффективного и адаптивного захвата зависимостей различных размерностей (краткосрочных и долгосрочных). У данной сети есть гейты обновления и сброса. Первый отвечает за выбор того, какую часть знаний следует сохранить, второй находится между двумя последовательными повторяющимися единицами и решает, какой объем информации необходимо забыть.

Гейт обновления GRU можно описать как:

$$\mathrm{z}_t^j = \sigma\left(\mathbf{W}_\mathrm{z}\mathbf{X}_t + \mathbf{U}_\mathrm{z}\,\mathrm{H}_{t-1}\right)^j, \tag{2.75}$$

где $\mathbf{W}_\mathrm{z}$ – вектор веса гейта обновления; $\mathbf{X}_t$ – входной вектор; $\mathbf{U}_\mathrm{z}$ – вектор веса гейта обновления; H_{t-1} – скрытое состояние предыдущей ячейки.

Гейт сброса можно описать как:

$$\mathrm{r}_t^j = \sigma\left(\mathbf{W}_r\mathbf{X}_t + \mathbf{U}_r\,\mathrm{H}_{t-1}\right)^j, \tag{2.76}$$

где $\mathbf{W}_r$ – вектор веса гейта сброса; $\mathbf{U}_r$ – вектор веса гейта обновления.

Тогда обновление состояния скрытого слоя на основе текущего входа и предыдущего состояния формализуется как:

$$\tilde{\mathrm{H}}_t^j = tanh\left(\mathbf{W}_{\mathrm{x}_t} + \mathbf{U}\left(\mathrm{r}_t \otimes H_{t-1}\right)\right)^j, \tag{2.77}$$

где $\mathbf{W}_{\mathrm{x}_t}$ – вектор гейта кандидата.

Активация GRU в момент *t* может быть формализована:

$$h_t^j = \left(1 - \mathrm{z}_t^j\right)\mathrm{H}_{t-1}^j + \mathrm{z}_t^j \times \tilde{\mathrm{H}}_t^j. \tag{2.78}$$

Среди преимуществ GRU можно выделить вычислительную эффективность, состоящую в меньших в сравнении с LSTM требованиях к ресурсам за счет меньшего числа внутренних параметров, не сказывающегося негативно на эффектив-

ности сети. Преимущества LSTM заключаются в лучшем понимании долгосрочных зависимостей и в устойчивости к затуханию градиента.

Простые RNN и их глубокие вариации по своей конструкции зависят от порядка вхождения обрабатываемой последовательности. Изменение порядка влияет на способность модели к интерпретации.

A. Graves и C. Schmidhuber [93] разработали новую технику извлечения контекстных зависимостей между словами в текстовых последовательностях. Идея заключалась в том, чтобы ввести дополнительный уровень, который будет обрабатывать входные данные в обратном порядке по сравнению с первым слоем, что приведет к анализу последовательности как слева направо, так и справа налево. За счет **обработки данных в двух противоположных направлениях** модель способна находить признаки, которые были бы упущены при традиционном одностороннем хронологическом обходе.

Еще одно решение на базе моделей глубокого обучения подразумевает объединение рекуррентных и сверточных архитектур в сверточно-рекуррентную сеть (**CRNN**) с целью выявления большего числа информативных признаков как явного, так и неявного характера. Слои CRNN могут варьироваться, однако сохранять общую структуру данной сети. Как правило, она состоит из нескольких сверточных слоев разных размерностей, позволяющих находить признаки различного масштаба, простого рекуррентного слоя или двунаправленного, а также полносвязного слоя для объединения информации и решения задачи классификации (или другой).

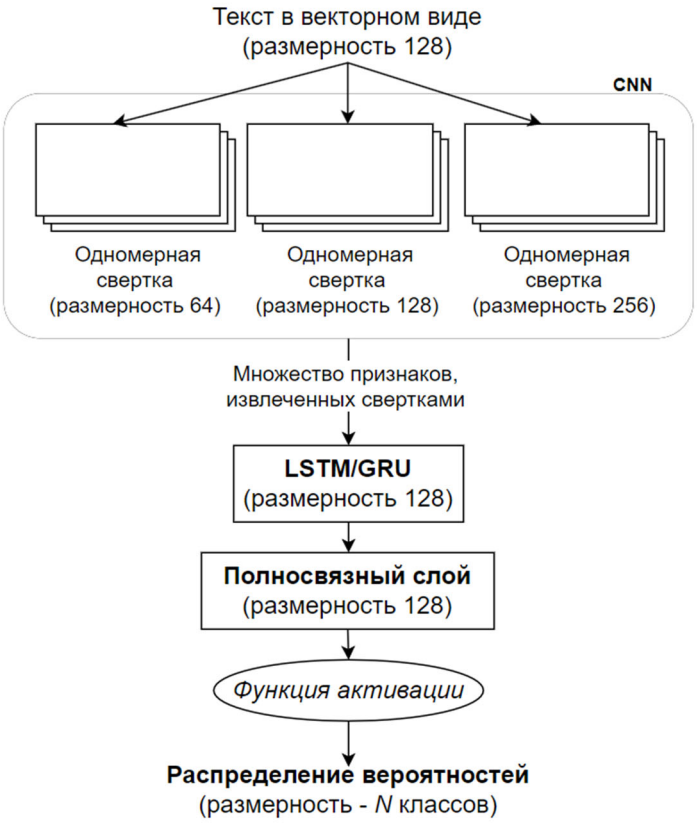


Рисунок 2.14 – Обобщенная архитектура CRNN

На рисунке 2.14 представлена обобщенная архитектура CRNN. Входной слой принимает на вход векторы размерностью не более 128. Затем осуществляется параллельная работа одномерных сверток размерностями 64, 128 и 256, позволяющих выявить локальные и глобальные информативные признаки. Результаты сверток конкатенируются по завершении их анализа. На вход рекуррентной сети передаются признаки, извлеченные сверточной частью. Работу завершает полносвязный слой и функция активации, позволяющие получить итоговое распределение вероятностей классификатора CRNN.

В 2017 году была представлена модель под названием «Attention is All You Need» [18], которая предложила новый подход к обработке последовательностей с использованием механизма внимания. Эта модель стала основой для трансформеров. Архитектура базовой трансформерной модели представлена на рисунке 2.15. Она состоит из двух основных компонентов: энкодера и декодера. Каждый из них содержит несколько слоев самовнимания (self-attention) и полносвязанных слоев (feed-forward). Работу трансформеров можно разбить на этапы.

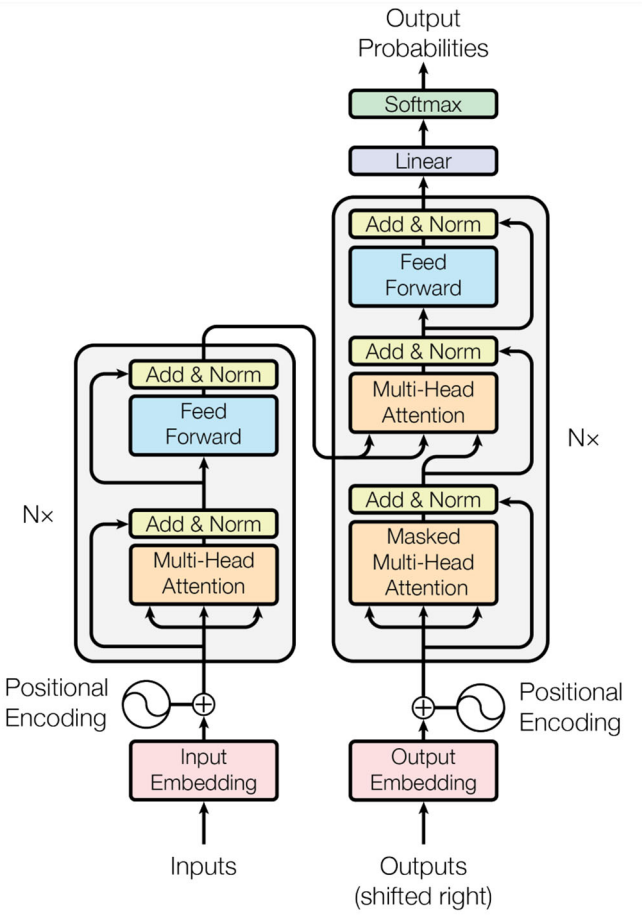


Рисунок 2.15 – Архитектура трансформера

1. Самовнимание. В контексте «Attention is all you need» используется механизм самовнимания мульти-хэд (multi-head attention), позволяющий обрабатывать различные аспекты входных данных одновременно, что способствует изучению более сложных зависимостей и представлений.

2. Входные данные подвергаются линейным преобразованиям для получения трех наборов весов: запросы (**Q**), ключи (**K**) и значения (**V**).

3. Каждый набор весов используется для вычисления отдельного внимания (attention) с помощью обычного самовнимания.

4. Результаты этих отдельных вниманий объединяются и подвергаются дополнительному линейному преобразованию, чтобы получить окончательный вывод мульти-хэд внимания.

Формально данные операции можно описать как:

$$Attention\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=softmax\left(\frac{\mathbf{QK}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V}, \tag{2.79}$$

$$head_i=Attention\left(\mathbf{QW}_i^{\mathbf{Q}},\mathbf{KW}_i^{\mathbf{K}},\mathbf{VW}_i^{\mathbf{V}}\right), \tag{2.80}$$

$$MultiHead\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=Concat\left(head_1,...,head_h\right)\mathbf{W}^{\mathbf{O}}, \tag{2.81}$$

где проекции представляют собой матрицы параметров $\mathbf{W}_{\mathrm{i}}^{\mathbf{Q}}\in\mathbf{R}^{d_{model}\times d_k}$, $\mathbf{W}_{\mathrm{i}}^{\mathbf{K}}\in\mathbf{R}^{d_{model}\times d_k}$, $\mathbf{W}_{\mathrm{i}}^{\mathbf{V}}\in\mathbf{R}^{d_{model}\times d_V}$, $\mathbf{W}^{\mathbf{O}}\in\mathbf{R}^{hd_v\times d_{model}}$; h – число слоев внимания; d_{model} – размерность модели.

Авторы оригинальной статьи используют 8 параллельных слоев внимания или «голов». Для каждого из них рассчитываются $d_k=d_v=d_{model}/h=512/8=64$. Из-за уменьшенного размера каждой головы общие вычислительные затраты аналогичны затратам на внимание одной головы с полной размерностью.

5. Полносвязные слои. Слои полной связи выполняют нелинейные преобразования признаков.

6. Многослойность. Архитектура трансформера содержит несколько слоев энкодера и декодера, что позволяет модели изучать более сложные зависимости и представления входных данных.

7. Позиционное кодирование. Для того чтобы модель могла учитывать порядок элементов в последовательности, в трансформере используется позиционное кодирование, которое добавляет информацию о позиции каждого элемента.

8. Обучение с учителем. В декодере используется метод обучения с учителем, при котором на каждом шаге модель получает на вход предыдущие выходы.

Трансформерная модель легла в основу многих современных архитектур, среди которых можно выделить GPT и BERT. Модель **GPT** была впервые представлена в статье [180] командой исследователей OpenAI. Эта модель представляет собой пример архитектуры, способной генерировать текст и выполнять задачи обработки естественного языка. GPT (рисунок 2.16) использует те же компоненты, что и трансформер, такие как механизм внимания и многослойные НС. GPT обучается на больших объемах текстовых данных с целью понимания контекста и генерации качественных текстовых ответов. Кроме того, GPT использует методы предварительного обучения и дообучения, чтобы достичь высокой производительности и адаптироваться к конкретным задачам обработки естественного языка.

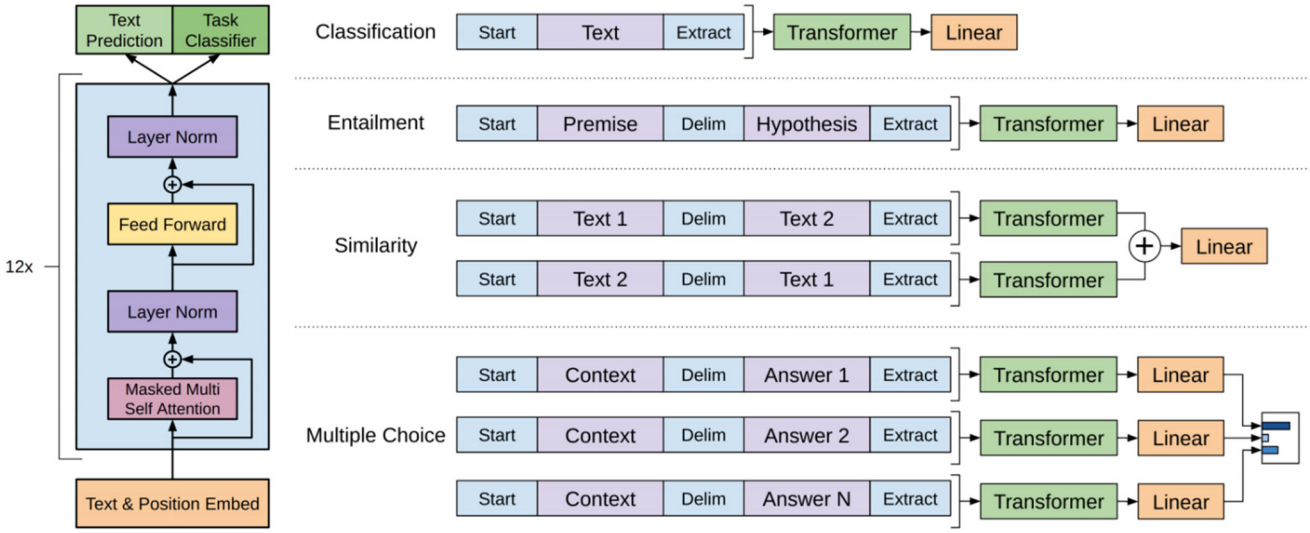


Рисунок 2.16 – Архитектура GPT

Одной из наиболее известных стала архитектура **BERT** [30]. BERT представляет собой модель глубокого обучения, способную понимать контекст и взаимосвязь между словами в предложении. В отличии от трансформерной модели, BERT использует только механизм кодирования. Также в отличии от GPT данная модель использует двунаправленный подход и обрабатывает контекст слева и справа.

Основной принцип работы BERT заключается в предобучении модели на большом объеме текстовых данных, чтобы она могла усвоить общее понимание о языке. В процессе предобучения BERT использует два основных подхода:

1. Маскирование токенов: BERT случайным образом маскирует некоторые слова в предложении и требует от модели предсказать их исходные значения. Это позволяет модели учиться понимать контекст и связи между словами.

2. Предсказание следующего предложения: BERT обучается предсказывать, является ли одно предложение следующим за другим в тексте. Это помогает модели учиться улавливать связи между предложениями и контекстом.

Глубокие модели на базе трансформеров имеют несколько преимуществ по сравнению с классическими нейросетями и методами машинного обучения:

1. Умение улавливать долгосрочные зависимости: трансформеры способны обрабатывать последовательности данных, такие как тексты, с учётом долгосрочных зависимостей между элементами последовательности.

2. Параллельная обработка: трансформеры могут обрабатывать данные параллельно благодаря механизму внимания, что ускоряет процесс обучения.

3. Обучение на больших объемах данных: трансформеры успешно применяются для предобучения на больших объемах текстовых данных, что позволяет им усваивать общее понимание о языке и контексте.

4. Обобщение на различные задачи: после предобучения трансформеры могут быть дообучены под различные задачи без необходимости переобучения.

Сиамские НС (SimNN) [132, 154] представляют собой специальный тип НС, предназначенный для решения задач сопоставления и подтверждения. Их особенностью является наличие «сиамских» ветвей, имеющих одинаковую архитектуру и веса. Принцип работы SimNN заключается в том, что данные подаются на вход каждой из «сиамских» ветвей сети, которые затем обрабатывают эти данные и извлекают их признаки. Затем признаки, полученные из каждой ветви, объединяются и проходят через дополнительные слои модели для сравнения или классификации. Важно отметить, что в качестве ветвей SimNN могут выступать любые другие архитектуры, например, CNN, RNN, Transformer и их вариации.

Подводя итог, можно сделать следующие выводы о методах решения задач интеллектуального анализа текста:

– к положительным качествам классических НС и методов машинного обучения можно отнести возможность интерпретации полученных результатов, простоту реализации в совокупности с высокой вычислительной эффективностью и способности работать с небольшими корпусами.

– поскольку различные типы RNN могут использовать для обработки последовательностей произвольной длины свою внутреннюю память, это позволяет им достигать наибольшей эффективности в задачах анализа текста. Это относится к таким типам RNN, как LSTM и GRU, решающим проблему исчезновения или взрыва градиентов методом обратного распространения ошибки;

– гибридные НС способны устранять недостатки друг друга за счет комбинирования двух и более НС. Объединение LSTM/GRU, имеющей множество параметров обучения, с CNN, использующей сравнительно небольшое количество внутренних параметров, может сократить время обучения модели. Двунаправленные RNN лучше извлекают и позволяют сохранить контекст в длинных текстовых последовательностях, а CNN фокусируется на локальных последовательностях токенов разных размерностей, что обеспечивает высокую точность классификации.

– добавление механизма внимания может повысить качество анализа текста за счет учета закономерностей между токенами, распределенными по всему тексту.

2.8 Используемые методы оптимизации гиперпараметров

Как методы машинного обучения, так и методы на основе НС в большинстве случаев оснащены рядом гиперпараметров (задающиеся до обучения модели). Их настройка напрямую влияет на конечный результат:

– подбор оптимальных значений гиперпараметров может значительно улучшить производительность модели, позволяя ей лучше адаптироваться к данным и делать более точные прогнозы;

– неправильно подобранные гиперпараметры могут привести к переобучению (слишком сложная модель) или недообучению (слишком простая модель), что отрицательно скажется на способности модели обобщать данные;

– подбор оптимальных гиперпараметров позволяет избежать излишнего использования вычислительных ресурсов на обучение и тестирование моделей с неподходящими параметрами;

– правильно подобранные гиперпараметры могут сделать модель более устойчивой к изменениям в данных, что важно при решении реальных задач.

Оптимизация гиперпараметров вручную ресурсозатратна и не всегда эффективна, поэтому ее следует осуществлять с использованием специальных методов оптимизации. Среди наиболее распространенных из них можно выделить методы жадного поиска, лучевого поиска и эвристические.

Жадный поиск предполагает перебор всех возможных комбинаций значений гиперпараметров из заданного диапазона. Выбирается та комбинация, которая дает наилучшие оценки на валидационной выборке.

Лучевой поиск основан на переборе комбинаций значений гиперпараметров, но в отличие от жадного поиска, он выбирает случайные комбинации для оценки. Лучевой поиск (рисунок 2.17) предполагает оптимизацию по принципу «наилучшее первое», которая снижает требования к памяти. Это поиск по графу, который упорядочивает все состояния в соответствии с выбранной эвристикой. При лучевом поиске в качестве кандидатов сохраняется только заранее определенное количество

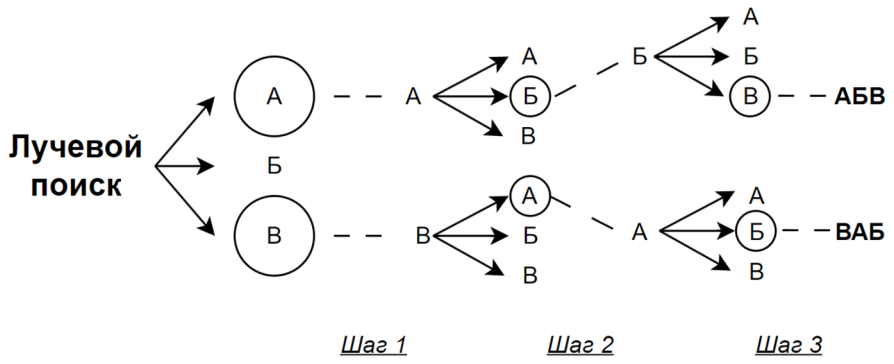


Рисунок 2.17 – Пример работы лучевого поиска

лучших частичных решений, поэтому лучевой поиск считает частным случаем жадного.

Недостатком метода является отсутствие гарантии, что лучшее решение будет обнаружено. Преимуществами: уменьшение количества комбинаций, которые необходимо оценить по сравнению с жадным поиском и возможность обнаружения оптимальных значений гиперпараметров.

Эмпирический метод основан на использовании статистических методов для оптимизации гиперпараметров. Он стремится найти оптимальные значения гиперпараметров, используя информацию о предыдущих оценках модели. Некоторые из таких статистических методов включают:

1. Гауссовский процесс – это вероятностная модель, которая используется для моделирования неизвестной функции. Он может быть применен для моделирования функции оценки гиперпараметров и предсказания оптимальных значений гиперпараметров на основе предыдущих оценок.

2. Адаптивные методы оптимизации – методы, использующие информацию о предыдущих оценках для адаптивного выбора следующей точки для оценки. Примеры таких методов включают адаптивную выборку, которая учитывает неопределенность в моделировании функции оценки.

3. Байесовские методы – методы, применяемые для инкорпорации априорной информации о распределении гиперпараметров и их взаимосвязях. Они позволяют эффективно исследовать пространство гиперпараметров и сокращать количество необходимых оценок модели.

2.9 Используемые методы снижения размерности

Отбор признаков часто используется как этап предварительной обработки данных [156, 165, 240, 252, 291] для последующего применения методов машинного обучения. Он представляет собой процесс формирования подмножества признаков таким образом, чтобы оно было наиболее информативным для алгоритмов.

Это приводит к исключению ненужных и избыточных признаков, повышения точности алгоритмов, ускорении процесса обучения.

Алгоритмы отбора информативных признаков можно разбить на две категории: модели на основе фильтрации и оберточные модели. Первые опираются на общие характеристики данных для выбора некоторых признаков без использования какого-либо алгоритма обучения. Оберточные модели требуют одного заранее определенного алгоритма обучения для его использования в процессе отбора признаков. Оберточные модели имеют тенденцию находить признаки, лучше подходящие для заранее определенного алгоритма обучения, что приводит к высокой эффективности обучения алгоритмов. Тем не менее, если число признаков слишком велико, применение моделей на основе фильтров может быть целесообразным.

2.9.1 Методы на основе фильтрации

Наиболее популярными моделями на основе фильтрации являются дисперсионный анализ (ANOVA), метод главных компонент (PCA), фильтр быстрой корреляции (FCBF).

ANOVA – статистический метод, используемый для сравнения средних значений трех или более групп и определения наличия статистически значимых различий между ними, был впервые упомянут Р. Фишером [79]. ANOVA разделяет общую изменчивость данных на две части: изменчивость между группами и изменчивость внутри группы. Путем вычисления *f*-статистики ANOVA определяет отношение межгрупповой изменчивости к изменчивости внутри группы. Если значение *f*-статистики превышает критическое значение, это указывает на статистически значимые различия между средними значениями групп.

Анализ ANOVA для фильтрации признаков выполняется в несколько шагов:

1. Выполнить однофакторный ANOVA: рассчитать *f*-статистику для каждого признака, используя зависимую переменную как группирующий фактор:

$$f = \frac{MST}{MSE}, \tag{2.82}$$

где *MST* – средняя сумма квадратов вследствие обработки; *MSE* – средняя сумма квадратов вследствие ошибки.

2. Определить, является ли значение *f*-статистики статистически значимым с учетом выбранного уровня значимости (обычно 0,05).

3. Отобрать признаки, для которых значение *f*-статистики превышает уровень значимости. Это означает, что эти признаки имеют статистически значимое влияние на зависимую переменную.

4. Удалить признаки, для которых значение *f*-статистики не превышает уровень значимости, так как они не вносят статистически значимого вклада в модель.

PCA [168] – метод уменьшения размерности признакового пространства, в котором итоговая матрица в порядке убывания содержит векторы с наибольшей информацией изначальных признаков. Для фильтрации признаков PCA осуществляются следующие шаги:

1. Стандартизация данных: перед применением PCA необходимо стандартизировать данные, чтобы убедиться, что все признаки имеют одинаковый масштаб.

2. Вычисление ковариационной матрицы, которая показывает связи между признаками.

$$Cov\left(X_i, X_j\right) = E\left[\left(X_i - E\left(X_i\right)\right)\cdot\left(X_j\right) - E\left(X_j\right)\right], \tag{2.83}$$

где $Cov\left(X_i, X_j\right)$ – матрица, у которой (*i*,*j*)-элемент является корреляцией признаков $\left(X_i, X_j\right)$.

3. Вычисление собственных значений и собственных векторов. Собственные векторы представляют новые направления (главные компоненты), а собственные значения показывают их важность.

4. Выбор компонент: на основе собственных значений выбираются наиболее значимые главные компоненты, которые содержат наибольшую часть информации изначальных признаков.

5. Преобразование данных с использованием выбранных главных компонент, что позволяет уменьшить размерность пространства признаков.

6. Анализ главных компонент: анализ новых компонент для понимания, какие именно признаки оказывают наибольшее влияние на изменение данных.

Принцип работы **FCBF** [77] заключается в вычислении корреляции между каждым признаком и целевой переменной, а затем в оценке также корреляции между самими признаками. После этого происходит отбор признаков на основе ранжирования их значимости. Работа FCBF начинается с полного множества признаков, определенных для анализа. Далее из него исключаются малоинформативные признаки. Для определения зависимостей между признаками FCBF использует меру симметричной неопределенности:

$$SU(X,Y) = 2\left[\frac{H(X) - H(X \mid Y)}{H(X) + H(Y)}\right] = SU(Y,X), \tag{2.84}$$

где $H(X)$, $H(Y)$ – энтропии случайных величин, имеющих соответственно i и j состояний; $H(X|Y)$ – условная энтропия:

$$H(X \mid Y) = -\sum_j P\left(y_j\right)\sum_i P\left(x_i \mid y_i\right)\log_2\left(P\left(x_i \mid y_j\right)\right), \tag{2.85}$$

где $P\left(x_i\right)$, $P\left(y_i\right)$ – априорные вероятности для всех значений X и Y, $P\left(x_i \mid y_j\right)$ – апостериорная вероятность X при известной Y.

Зависимость признаков считается тем ниже, чем ближе значение SU к нулю.

Среди рассмотренных моделей на основе фильтрации FCBF отличается скоростью и эффективностью отбора признаков, а также устойчивостью к шуму, однако не учитывает нелинейные зависимости между признаками и целевой переменной. PCA позволяет сократить размерность данных и уменьшить мультиколлинеарность, однако теряет интерпретируемость исходных признаков и является чувствительным к выбору числа компонент. ANOVA учитывает вариации между группами и подходит для категориальных переменных, однако является чувствительным в выбору модели и типам зависимостей.

2.9.2 Методы на основе оберточных моделей

Среди оберточных моделей можно выделить генетические алгоритмы (ГА), алгоритмы на основе регуляризации (RS), прямой последовательный отбор (FS), обратный последовательный отбор (BS), пояснения к аддитивам Шепли (SHAP).

Процесс отбора признаков с использованием **генетических алгоритмов** [78] может быть описан следующим образом:

1. Кодирование признаков: каждое решение (индивид) представляется в виде бинарной строки, где каждый бит соответствует наличию или отсутствию соответствующего признака в подмножестве.

2. Определение функции приспособленности: определяется метрика, например, качество модели, которая используется для оценки каждого индивида.

3. Инициализация популяции: случайным образом формируется начальная группа индивидуумов (подмножеств признаков) в популяции.

4. Основные генетические операторы: скрещивание (кроссовер): случайные пары индивидуумов объединяются для порождения новых потомков; мутация: случайные изменения битов в индивиде; отбор: выбор лучших индивидов для перехода в следующее поколение.

5. Оценка приспособленности: вычисление значения функции приспособленности для каждого индивида в популяции.

6. Цикл эволюции: повторение операций скрещивания, мутации, отбора и оценки приспособленности для создания новых поколений популяции.

7. Остановка: выполнение до достижения критерия останова, например, определенного количества поколений или сходимости решений.

Алгоритм отбора на основе RS [224] основан на присвоении меры информативности каждому признаку. Если она превышает величину порога или находится в допустимом диапазоне, то признак считается информативным и остается в модели. Процесс регуляризации заключается в добавлении штрафа к параметрам модели. Штраф позволяет избежать настройки модели под решение конкретно заданной задачи, а следовательно, ее неприменимости для других случаев. При *L1*-

регуляризации штраф применяется к коэффициентам, уменьшающим каждый из предикторов. Метод подходит для линейных классификаторов.

Метод, основанный на FS [4] представляет собой итеративный фильтрационный подход, который является простым способом уменьшения размерности признакового пространства. Заданное количество признаков подбирается путем полного перебора. На каждой итерации выбирается тот, что имеет лучшую эффективность. Начальный признак выбирается произвольно. Полный перебор может негативно отразиться на времени отбора, а также привести к переобучению модели из-за возможной сильной корреляции между отобранными признаками, несмотря на их близкое смысловое значение.

Алгоритм на основе BS [37] противоположен прямому. В первой итерации алгоритм работает на основе всех признаков. На каждой итерации исключается признак, не несущий ценности для модели. В этом методе тоже присутствует неопределенность при удалении сильно коррелированных признаков.

Алгоритм SHAP, предложенный S. Lundberg и L. Su-In в 2017 г. [138] – использует идеи из теории игр. Метод вычисляет, какой вклад каждый игрок (признак) внес в итоговый результат, используя специальный подход, называемый значением Шепли. Формально это отражено в следующем виде:

$$g\left(z'\right) = \varphi_0 + \sum_{j=1}^{\|\mathbf{M}\|} \varphi_j z_j , \tag{2.86}$$

где g – используемая линейная модель; $z' \in [0;1]$, $\|\mathbf{M}\|$ – длина вектора коалиции (объединения игроков), в котором запись 1 означает, что соответствующее значение признака присутствует, а 0 – отсутствует; $\varphi_j \in R$ – значения Шепли, выступающие как атрибутивная функция для j-го признака, 0 – значение Шепли до включения j-го признака. Условие об использовании линейной модели обусловлено возможностью вычисления φ.

Метод может работать с обученной моделью и определять информативность признаков на основе оценки тестовых экземпляров, в то время как другие методы определяют информативность в процессе обучения модели.

2.10 Используемые методы сглаживания

При использовании *n*-грамм существует риск получения «разреженных данных». Вероятность их появления тем меньше, чем меньше размер текста и чем больше длина *n*. Часть признаков могут отсутствовать вовсе. Решением является использование методов сглаживания, подробно описанных в работе [49], позволяющих оценить вероятности малочастотных *n*-грамм.

Метод Гуда – Тьюринга базируется на подсчете N_C – количества *n*-грамм, встретившихся ровно C раз. Частота C заменяется дисконтированной оценкой Гуда – Тьюринга C^*, как функцией от величины N_{k+1}. Для больших значений, когда $C>k$ (где k – некоторый порог), значения C принимаются правдоподобными. Катц [120] предложил величину порога, равную 5. Итоговое выражение для вычисления C^* записывается как:

$$C^* = \begin{cases} C, & \text{если } C > k \\ \dfrac{(C+1)\dfrac{N_{C+1}}{N_C} - C\dfrac{(k+1)N_{k+1}}{N_1}}{1 - \dfrac{(k+1)N_{k+1}}{N_1}}, & \text{если } 1 \le C \le k, \end{cases} \tag{2.87}$$

где N_1 – число слов, встречающихся в тексте один раз, *N*-граммы с малой частотой встречаемости обычно обрабатываются также, как в случае $N_C = 0$.

В **методе Катца** дисконтированная вероятность P^* используется для уменьшения оценок максимального правдоподобия вероятностей и сохранения части вероятностной меры для *N*-грамм меньшего порядка:

$$P_{KATZ}\left(a_i \mid a_{i-n+1},...,a_{i-1}\right)=$$

$$=\begin{cases} P^*\left(a_i \mid a_{i-n+1},....,a_{i-1}\right), \text{ если } C\left(a_{i-n+1},...,a_i\right)>k \\ \dfrac{1-\sum_{a_i:C(a_{i-n+1},...,a_i)>0} P^*\left(a_i \mid a_{i-n+1},...,a_{i-1}\right)}{1-\sum_{a_i:C(a_{i-n+1},...,a_i)>0} P^*\left(a_i \mid a_{i-n+2},...,a_{i-1}\right)} P_{KATZ}\left(a_i \mid a_{i-n+2},...,a_{i-1}\right), \\ \text{если } 1\le C\left(a_{i-n+1},...,a_i\right)\le k. \end{cases} \quad (2.88)$$

2.11 Различие естественно- и искусственно-языковых текстов

Естественно-языковые тексты (художественные произведения, любительские тексты, сообщения и др.) и искусственно-языковые тексты (исходные коды программ на различных языках программирования) существенно отличаются друг от друга по ряду признаков.

Исходные коды программ строго структурированы и следуют точным синтаксическим и семантическим правилам языка программирования, определяющим функциональные требования к программе. Каждая инструкция, переменная, функция, класс и т.д., должны быть оформлены в соответствии с правилами языка, иначе программа не будет работать. Все конструкции языка должны однозначно пониматься транслятором, осуществляющим преобразование программы, написанной на высокоуровневом языке программирования, в машинный код. Семантика искусственно-языковых текстов определена очень строго для каждой отдельной конструкции языка программирования. Любые нарушающие ее изменения, вносимые автором-программистом, могут привести к нарушению работоспособности программы. В связи с этим, авторские признаки искусственно-языковых текстов могут быть значительно менее выраженными, чем в естественно-языковых.

Естественный текст гораздо более гибок в плане структуры и синтаксиса. Хотя существуют правила грамматики и пунктуации, в естественных языках допу-

стимо большее разнообразие выражений и стилей. Синтаксис может являться предметом творческого выражения автора. Семантика естественно-языковых текстов может быть многозначна ввиду зависимости от контекста. Такие тексты имеют сложную структуру, которая может состоять из различных тропов, таких как метафоры, аллегории, перифразы, а также неоднозначные кореферентности, требующие их разрешения для точного понимания семантики текста. Кроме того, естественно-языковые тексты сильно зависимы от множества человеческих и внешних факторов. Например, настроение автора, понимание им предмета обсуждения, уровень его образования и квалификации, а также взгляды могут оказывать существенное влияние на стиль и структуру текста. Аналогичным образом на естественно-языковой текст влияет и внешняя среда, например, необходимость соответствия определенной целевой аудитории, правилам, цензуре.

Таким образом, структура исходных кодов программ более формализована и ориентирована на создание функциональности программного обеспечения, в то время как естественный текст обладает большей гибкостью и используется для передачи информации и идей. Это обуславливает необходимость разработки отдельных методик анализа, учитывающих особенности естественно- и искусственно-языковых текстов, как объектов исследования.

2.12 Выводы по главе

Подводя итог исследования, выделим следующие основные результаты:

1. На основе анализа материалов по теме диссертации, рассмотренных в первой главе, предложена методология идентификации автора естественно-языкового текста и исходных текстов программ для решения задач кибербезопасности в виде комплекса методов, моделей и алгоритмов, агрегирующий имеющийся опыт. Методология является универсальной для решения задач кибербезопасности, связанных с классификацией текстов.

2. Предложена модель создания автором текста в киберсреде, учитывающая взаимодействие компонентов и ограничения, накладываемые на процесс личностью автора и видом деятельности, особенностями среды, текста и семантики.

3. Методология предполагает использование методов статистического анализа, машинного и глубокого обучения. Исходя из специфики данных, решаемой задачи кибербезопасности и потенциальных атак на метод, на каждом из этапов могут быть задействованы разные по принципу действия подходы. Традиционные методы машинного обучения обеспечивают высокую степень интерпретируемости и скорости, поэтому могут применяться как самостоятельно, так и в составе ансамблей методов принятия решений, однако являются менее эффективными в сравнении с НС при наличии осложняющих факторов. НС являются более устойчивыми к атакам на метод и более эффективными для поиска явных и неявных признаков авторского стиля.

4. Ключевыми в методологии являются методики идентификации автора естественно-язычного и искусственно-язычного текста, которые требуют отдельного рассмотрения и апробации, так как подразумевают применение разных по своему принципу действия методов и инструментов. Это связано со спецификой анализируемых данных. Подходы, используемые для естественно-языкового анализа, должны обеспечивать интерпретацию неоднозначности, понимание контекста, а также разрешение двусмысленности, которые являются менее характерными для искусственно-языковых текстов. Подходы, используемые для искусственного языка, напротив, должны быть адаптированы под анализ регулярных структур, информативные признаки в которых могут быть менее выраженными ввиду следования авторами строгим синтаксическим и семантическим инструкциям и правилам.

3 МЕТОДИКА ИДЕНТИФИКАЦИИ АВТОРА ЕСТЕСТВЕННО-ЯЗЫКОВОГО ТЕКСТА

3.1 Закрытая и открытая атрибуция, верификация автора

Идентификацию автора текста [20, 21, 281, 294, 298] рассматривают в двух случаях: закрытое и открытое множество авторов. Различие заключается в том, что в случае открытого множества кандидатов истинного автора может вовсе не быть в списке кандидатов, в то время как в случае закрытого множества он обязательно присутствует.

Методы определения автора применимы при решении актуальных прикладных задач в области защиты интеллектуальной собственности и охраны авторских прав на научные и художественные текстовые произведения, выявления распространителей противозаконной информации в социальных сетях, улучшения качества образовательного процесса и повышения оригинальности научных трудов.

Верификация авторства может использоваться для продленной аутентификации по тексту в социальных сетях и мессенджерах.

При математической постановке задачи **закрытой атрибуции** процесс установления авторства текста задан тремя конечными множествами: $\mathbf{A}=\{a_1,...,a_n\}$, $\mathbf{T}'=\{t_1,...,t_m\}$, $\mathbf{T}=\{t_1,...,t_s\}$ – авторов, текстов с известным авторством и анонимных текстов, соответственно. Каждому тексту, включая анонимные, соответствует вектор признаков.

Решение задачи сводится к вычислению целевой функции:

$$f(t_i)=\left[p(a_1),p(a_2),...,p(a_n)\right], \tag{3.1}$$

где t_i – произвольный текст; $p(a_1),p(a_2),...,p(a_n)$ – распределение вероятностей принадлежности текста авторам на основе вычисленных признаков текста. Конечным ответом в вопросе разрешения авторства по анонимному тексту является выбор автора с максимальным значением вероятности $\max(p_i)$.

Верификация авторства позволяет подтвердить или опровергнуть предположение о том, что тексты написаны одним и тем же автором. В отличие от идентификации, верификация предполагает использование только образцов $\mathbf{T}_a$ целевого класса a. Решение задачи сводится к построению бинарного классификатора, определяющего относится ли образец *t* к классу a или отрицательному классу $\overline{a}$:

$$verification(t,a) \to \{0,1\}, \tag{3.2}$$

где 0 – соответствует классу $\overline{a}$; 1 – соответствует классу a.

В случае возможного отсутствия истинного автора в списке претендентов, решается задача **открытой атрибуции**. Учитывается дополнительный нулевой класс a_{null}. Для того, чтобы его учесть можно использовать несколько подходов:

1) включить в образцы класса тексты как можно большего числа авторов $\mathbf{T}_{a_{null}} = \left\{t_1,...,t_{a_{null}}\right\}$. Но, должны учитываться ограничения $\mathbf{T}_{a_{null}} \cap \mathbf{T}' = \varnothing$, $\left|\mathbf{T}_{a_{null}}\right| \approx \underset{i}{avg}\left(\left|\mathbf{T}_{a_i}\right|\right)$, $a_i \in A$, чтобы нулевой класс не включал образцы известных классов и обучающая выборка была сбалансирована по количеству образцов. В случае ограниченного количества образцов каждого автора этот подход не применим, потому что количество образцов нулевого класса превышает обучающее $\left|\mathbf{T}_{a_{null}}\right| >> \left|\mathbf{T}'\right|$;

2) использовать *n* одноклассовых классификаторов, решающих задачу верификации для каждого потенциального автора. Решение отнести текст к нулевому классу формируется на основе *n* отрицательных решений точных одноклассовых классификаторов:

$$\sum_{j}^{n} verification\left(t,a_j\right) = 0 \Rightarrow t \in \mathbf{T}_{a_{null}}, \tag{3.3}$$

иначе решение о принадлежности образца известному классу принимается классификатором, обученным на известных текстах потенциальных авторов.

3.2 Методика идентификации автора для случая закрытой атрибуции

На сегодняшний день представлено множество исследований, подтверждающих эффективность SVM для классификации текста и определения авторства, в частности [66, 105, 185, 211, 299]. Появление сверточных НС и LSTM, GRU [8, 121] позволило улучшить полученные ранее показатели благодаря способности таких моделей обучаться на информативных характеристиках без предварительного их выделения как в случае с классическими методами машинного обучения. Методика закрытой атрибуции представлена на рисунок 3.1.

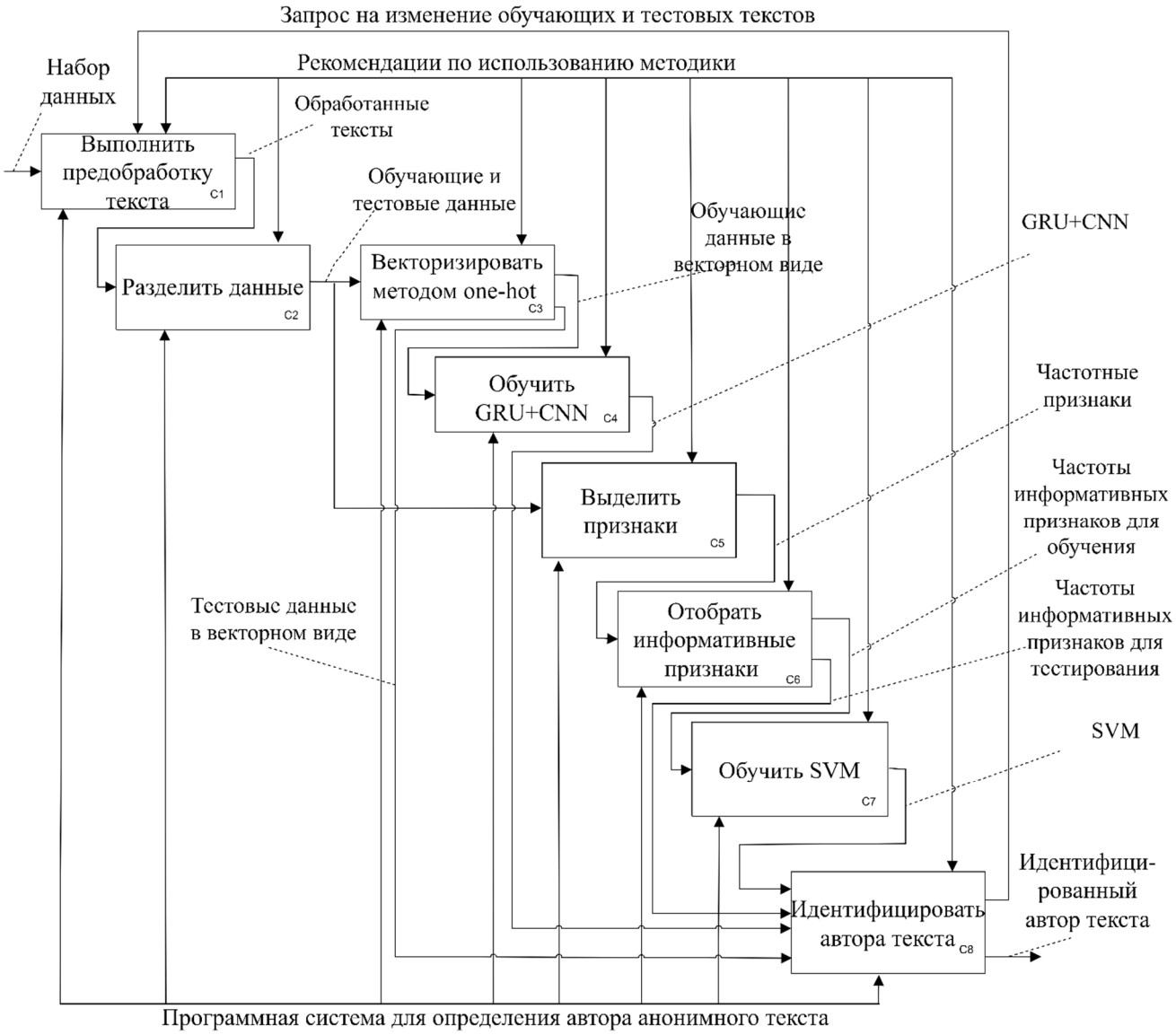


Рисунок 3.1 – Методика определения автора текста (закрытая атрибуция)

3.2.1 Предварительная обработка данных

На подготовительном этапе выполняется предварительная обработка данных, позволяющая исключить нерелевантные фрагменты текстов, шумы и привести образцы к единому формату. Предобработка включает приведение к единому регистру, очистку цифр и специальных символов, унификацию пунктуации.

3.2.2 Формирование признакового пространства в задаче идентификации автора текста

Признаковое пространство текста представляет собой вектор, составленный из значений, соответствующих некоторому набору признаков. По сформированному признаковому пространству классификатор способен выявить общие зависимости и взаимосвязи, присущие всем объектам (текстам), относящимся к данному классу (автору), в том числе тем, которые еще не наблюдались.

В последние десятилетия при исследовании языка конкретного автора особо актуально составление частотных словарей и анализ полученных данных. Материалы частотных словарей активно используются в решении множества задач, связанных с атрибуцией текста [75, 40], криминалистикой [184], психолингвистикой [108]. Статистическая характеристика текстов является математическим описанием детального изучения языка произведений писателя.

Для использования классических методов машинного обучения приоритетно определение признакового пространства в связи с неспособностью классификаторов самостоятельно находить информативные признаки в тексте.

Вектор признаков текста, подаваемый на вход классификаторов, включает в себя признаки из следующего перечня:

1. Частоты встречаемости знаков пунктуации (15 знаков пунктуации).
2. Частоты встречаемости частей речи (20 частей речи).

3. Частоты униграмм символов (33 – по количеству букв в русском языке).

4. Частоты биграмм (1089) и триграмм (3000) символов.

5. Частоты 3000 наиболее популярных слов русского языка [276].

Для приведения признаков к общей шкале без потери информации о различии диапазонов используется минимаксная нормализация:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \tag{3.4}$$

где X' – новое значение; X – исходное значение; $X_{\min}$ – минимальное значение; $X_{\max}$ – максимальное значение.

Для работы с НС тексты кодируются с помощью метода one-hot encoding (см. п. 2.4.4). Данный метод используется для преобразования входных данных НС, классических методов МО, частотного и тематического моделирования.

3.2.3 Отбор информативных признаков

Параметры, по которым можно определить индивидуальный авторский стиль, могут быть разными: предпочитаемые слова, местные речевые особенности, длина предложений, использование оборотов речи, словарный запас. Однако изменение данных параметров приводит к изменению частотных характеристик текста и возникает вопрос в определении множества наиболее информативных признаков, а также исключении избыточных.

Формально, исходная модель использует $|\mathbf{F}| = 7151$ признаков текста. При решении практических задач не все признаки полезны, поэтому мощность конечного множества информативных признаков может оказаться меньше $|\mathbf{F}^{inf}| < |\mathbf{F}|$. Вектор признаков подобран в работе [289] и включает 1168 элементов, однако в работе не использовались методы отбора, поэтому есть потенциальные возможности улучшения результатов.

Отбор признаков осуществлялся методами со следующими параметрами:

− GA: размер популяции – 200; коэффициент скрещивания – 0,5; коэффициент мутации – 0,2; количество популяций – 10.

− FS и BS: направление отбора – прямое в случае прямого отбора и обратное в случае последовательного.

− RS: тип регуляризации – l1; двойственная форма задачи оптимизации (dual) – отключено; максимальное количество итераций – 1000.

− SHAP: алгоритм оценки значений Шепли – kernel.

Отбираемое количество признаков устанавливается при постановке задачи (50, 100, 200, 400 или 500).

3.2.4 Валидация модели

Для экспериментальной оценки методики применялись как классические методы машинного обучения (деревья решений, метод опорных векторов, логистическая регрессия, метод *k*-ближайших соседей, наивный байесовский классификатор), так и нейросетевые методы (fastText, RuBERT, MultiBERT, CNN, LSTM, BiLSTM, GRU, BiGRU).

Дополнительно используются многослойные гибридные архитектуры НС на основе совместного использования рекуррентных и сверточных слоев с разными размерами фильтров. Особенностью комбинированных методов является выявление более общих закономерностей в анализируемых данных.

Гибридные архитектуры способны демонстрировать более высокие результаты ввиду компенсации недостатков одной сетей преимуществами другой. Так, рассматривая гибридную нейронную сеть, состоящую из CNN и LSTM или CNN и GRU, можно сказать о ее частях следующее: CNN умеет хорошо извлекать локальную информацию, но плохо выражает контекстную информацию; в свою очередь, LSTM и GRU могут извлекать контекстные зависимости, что способствует повышению эффективности классификации, однако время обучения такой модели велико. Объединяя эти сети, можно возложить обязанность за извлечение локальных

особенностей текста на CNN, а LSTM и GRU сохранят временную информацию и извлечет контекстуальные зависимости текста. Такая гибридная сеть может дать результат лучше, чем если бы части этой сети работали отдельно. В данном исследовании рассмотрены комбинации LSTM + CNN, CNN + LSTM, GRU+CNN, CNN + GRU, BiGRU + CNN, CNN + BiGRU.

Все результаты получены в рамках 10-фолдовой кросс-валидации. Для обучения используется 3 образца по 15000 символов в случае художественных и любительских текстов и 40 текстов длиной до 250 символов для комментариев пользователей социальных сетей. При тестировании использовался 1 текст и 10 сообщений соответственно. Обучение и тестирование проводится на датасетах, описанных в п. 6.7.1.

Начальные параметры обучения были выбраны исходя из опыта исследователей в области анализа текстовых данных [17, 86, 196]. Окончательные значения подобраны с помощью алгоритма жадного поиска. Итоговые значения приведены ниже:

– CNN: embedding слой, размерность – 600; сверточный слой: размерность слоя – 512, размер ядра свертки – 3, функция активации – rectified linear unit (ReLU); dropout слой, коэффициент – 0,3; GlobalMaxPooling1D слой; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– CNN + CNN: embedding слой, размерность – 500; сверточный слой: размерность слоя – 4096, размер ядра свертки – 3, функция активации – ReLU; сверточный слой: размерность слоя – 1024, размер ядра свертки – 3, функция активации – ReLU; dropout слой, коэффициент – 0,3; GlobalMaxPooling1D слой; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– LSTM: embedding слой, размерность – 500; LSTM слой: размерность – 128, коэффициент dropout – 0,3, рекуррентный dropout коэффициент – 0,3; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– Bidirectional LSTM: embedding слой, размерность – 500; BiLSTM слой: размерность – 256; BiLSTM слой: размерность – 128; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– CNN+GRU/GRU+CNN (изменения порядка сверточного и GRU слоя в зависимости от архитектуры): embedding слой, размерность – 200; сверточный слой: размерность – 4096, размер ядра свертки – 3, функция активации – ReLU; dropout слой, коэффициент – 0,2; GlobalMaxPooling; преобразование выходного массива в трехмерную форму для подачи на GRU-слой; GRU слой: размерность – 128; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– LSTM + CNN/CNN + LSTM (изменение порядка сверточных и рекуррентных слоев в зависимости от архитектуры): embedding слой, размерность – 300; LSTM слой: размерность – 256, коэффициент dropout для входных нейронов – 0,3, рекуррентный dropout коэффициент – 0,3; сверточный слой: размерность – 1024, размер ядра свертки– 3, функция активации – ReLU; dropout слой, коэффициент – 0,2; MaxPooling; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– GRU: embedding слой, размерность – 200; GRU слой: размерность – 256; dropout слой, коэффициент – 0,3; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– BiGRU: embedding слой, размерность – 200; двунаправленный GRU слой: размерность – 256; двунаправленный GRU-слой: размерность – 128; полносвязный слой: размерность – количество классов, функция активации – SoftMax.

– fastText: длина *n*-граммы слов – 1; функция потерь – ova; число эпох – 100; коэффициент скорости обучения – 0,8; размерность – 500 (только для коротких текстов).

– RuBERT и MultiBERT: токенизаторы bert-base-multilingual-case для MultiBERT и rubert-basecased для RuBERT; функция активации – ReLU; коэффициент dropout – 0,1; скорость обучения – 0,00004.

– RF: количество деревьев – 25; глубина дерева – 4.

– SVM: алгоритм обучения – sequential optimization method; ядро – линейное; параметр регуляризации – 1; допустимый уровень ошибки – 0,00001; нормирование – включено; эвристика сжатия – включено.

– KNN: количество соседей – 25.

– LR: параметр регуляризации – 1; допуск по критериям останова – 0,0001; максимальное количество итераций – 100; алгоритм оптимизации – liblinear.

– DT: максимальная глубина дерева – 8; функция для определения качества разбиения – gini.

3.3 Результаты экспериментов по определению авторства в случае закрытого множества кандидатов

3.3.1 Художественные тексты

3.3.1.1 Классические методы машинного обучения

Рассматривались эксперименты, направленные на классификацию художественных текстов 2, 5, 10, 20 и 50 авторов. Использовались классические методы МО – SVM, LR, NB, DT, KNN и RF в двух вариациях: обучение на признаковом пространстве из работы [289] и TF-IDF представлении текста. В таблице 3.1 указаны полученные результаты и время обучения моделей для наиболее ресурсозатратного эксперимента – классификации 50 авторов.

Для объемных текстов, несмотря на ограничение в символах, использование классических методов МО, обученных на сформированном признаковом пространстве, эффективно. Объяснение полученных результатов состоит в достаточном для определения авторского инварианта объеме текстовых фрагментов, полноте множества признаков, выбранных для идентификации автора, обучении на экспериментально подобранных параметрах, а также способностью SVM хорошо работать с признаковым пространством большого размера за счет высокой гибкости при решении задач классификации различного уровня сложности, что, как следствие, приводит к уменьшению количества ошибок. Использование TF-IDF во всех случаях преимущества не дает.

Таблица 3.1 – Результаты определения автора художественного текста классическими методами машинного обучения

Метод	Количество авторов					Время (50 авторов), с.
	2	5	10	20	50	
Обучение на признаковом пространстве, точность (%)						
SVM	95,4±1,7	94,6±2,1	81,9±4,6	63,3±4,8	37,7±6,3	118
LR	95,1±4,4	92,8±3,1	74,2±5,1	61,9±5,1	34,7±7,1	102
NB	91,9±3,4	87,1±3,6	74,4±5,6	58,3±4,5	29,8±5,6	97
DT	94,1±1,1	91,4±2,3	84,2±4,4	52,2±2,3	17,8±2,7	89
RF	97,4±1,7	92,1±3,7	67,6±2,6	62,2±3,7	35,9±2,5	99
KNN	94,0±2,6	81,1±2,1	79,9±3,3	51,9±5,2	41,9±4,0	101
Обучение на TF-IDF представлении текста, точность (%)						
SVM	92,4±1,1	84,5±3,4	72,2±5,6	55,2±4,2	33,2±4,8	93
LR	93,2±2,3	82,2±4,3	68,7±5,5	52,1±4,8	27,7±3,3	76
NB	85,7±2,7	72,7±4,8	59,4±4,9	49,3±5,3	22,8±4,1	81
Метод	Количество авторов					Время (50 авторов), с.
	2	5	10	20	50	
DT	90,1±1,1	81,1±3,2	75,3±2,2	33,6±3,1	16,1±5,1	82
RF	92,5±2,2	88,5±3,1	70,6±1,1	59,0±3,4	31,4±4,1	83
KNN	86,9±3,3	78,0±2,6	71,4±4,9	57,4±2,6	40,2±3,4	82

3.3.1.2 Отбор признаков

Все пять описанных в п. 3.2.3 методов отбора применялись совместно с SVM с целью установления лучшего подмножества признаков и самого результативного метода отбора для художественных текстов. Выбор объясняется тем, что SVM превосходит другие классические методы МО до 13,5%, а также высокой скоростью обучения модели. Целью проведения эксперимента является улучшение точности классификации для 20 и 50 авторов. Результаты представлены в таблице 3.2.

Примеры отдельных графиков на рисунках 3.2–3.5 отражают влияние (ось абсцисс) конкретных признаков (ось ординат) на результаты классификации по 5 авторам с применением методов отбора для художественных текстов. Значение влияния разделяется для каждого класса в соответствии с цветовой меткой автора.

Рисунок 3.6 демонстрирует прирост точности при классификации совместно с SHAP.

Следует отметить, что RS выделил наиболее информативными исключительно частоты знаков пунктуации. Такие же признаки встречаются как информативные на графиках SHAP и ГА, в то время как по методам FS и BS знаки пунктуации не входят в топ-5 информативных признаков вовсе. Наименьший прирост даже для самых информативных признаков наблюдается по результатам метода FS, где в число признаков с наибольшим влиянием входят частоты частей речи, однако частоты речи встречаются и в методе ГА, но с существенно большим приростом.

Таблица 3.2 – Отбор признаков для художественных текстов

Алгоритм	Количество авторов					Время, с
	2	5	10	20	50	
Точность, % (кол-во подобранных признаков)						
ГА	98,8±0,8 (2783)	98,2±1,6 (2834)	92,3±2,0 (3097)	80,8±2,2 (4522)	56,6±3,2 (5821)	31553
RS	98,6±1,4 (2338)	98,1±1,5 (3049)	94,2±1,8 (4147)	88,3±2,5 (5263)	56,9±2,2 (5538)	34264
FS	93,4±3,0 (2600)	92,7±2,4 (2900)	82,5±3,0 (4600)	63,2±2,4 (5100)	40,4±1,8 (5300)	79420
BS	92,4±5,4 (2100)	92,2±4,0 (2500)	82,6±4,9 (3900)	59,1±2,2 (4800)	36,2±5,2 (5100)	73442
SHAP	98,2±4,9 (2430)	94,0±3,4 (2543)	88,5±3,6 (3642)	72,9±4,1 (4320)	42,1±4,7 (5433)	33543
1168 признаков [289]	95,4±1,7	94,6±2,1	81,9±4,6	63,3±4,8	37,7±6,3	118
Без отбора, TF-IDF	92,4±1,1 (8151)	84,5±3,4 (19523)	72,2±5,6 (37035)	55,2±4,2 (74648)	33,2±4,8 (157623)	93
Без отбора, 7151 признак	92,2±2,3	79,8±2,8	71,1±3,0	50,2±2,9	40,0±1,8	142

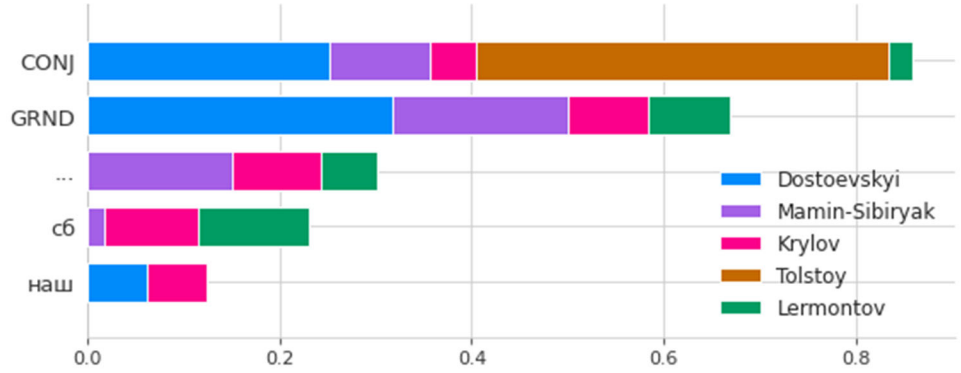


Рисунок 3.2 – Топ-5 информативных признаков, отобранных с помощью ГА

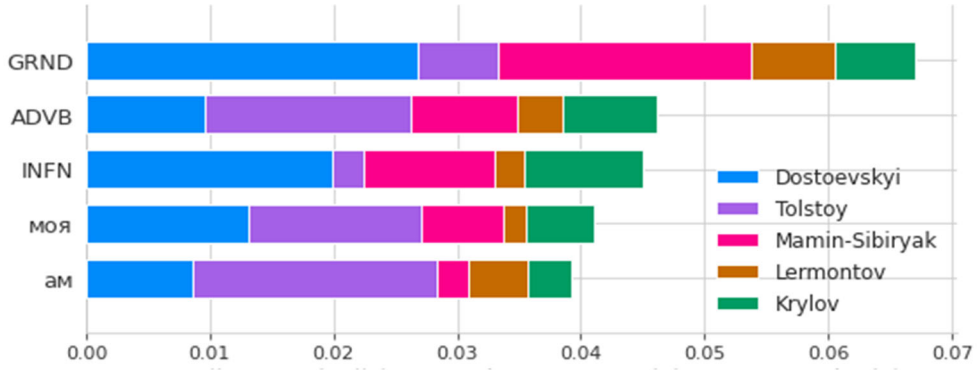


Рисунок 3.3 – Топ-5 информативных признаков, отобранных с помощью FS

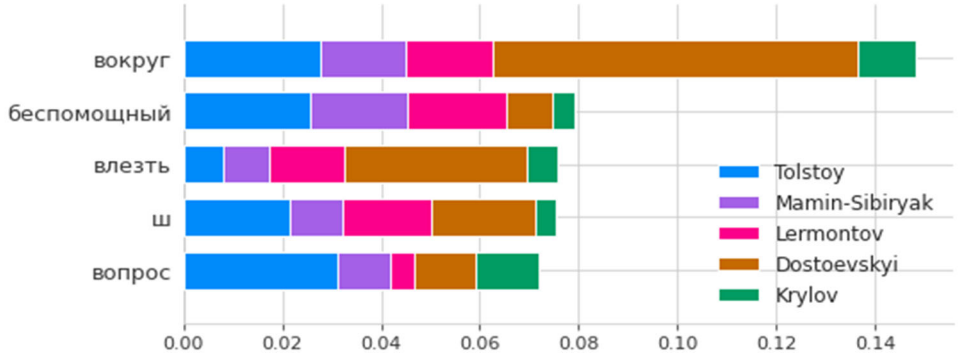


Рисунок 3.4 – Топ-5 информативных признаков, отобранных с помощью BS

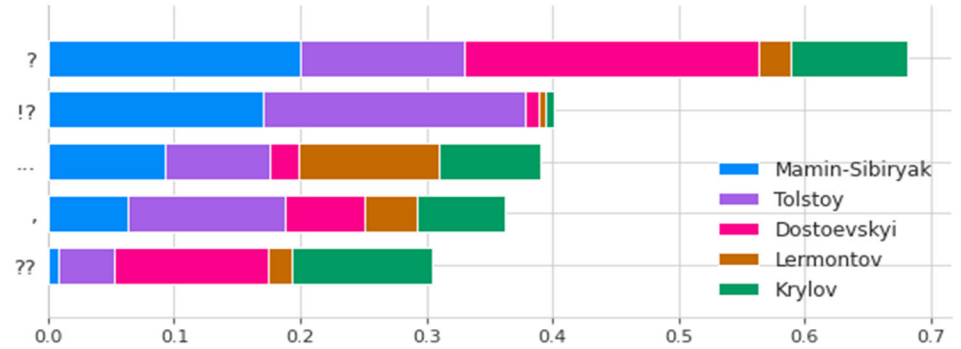


Рисунок 3.5 – Топ-5 информативных признаков, отобранных с помощью RS

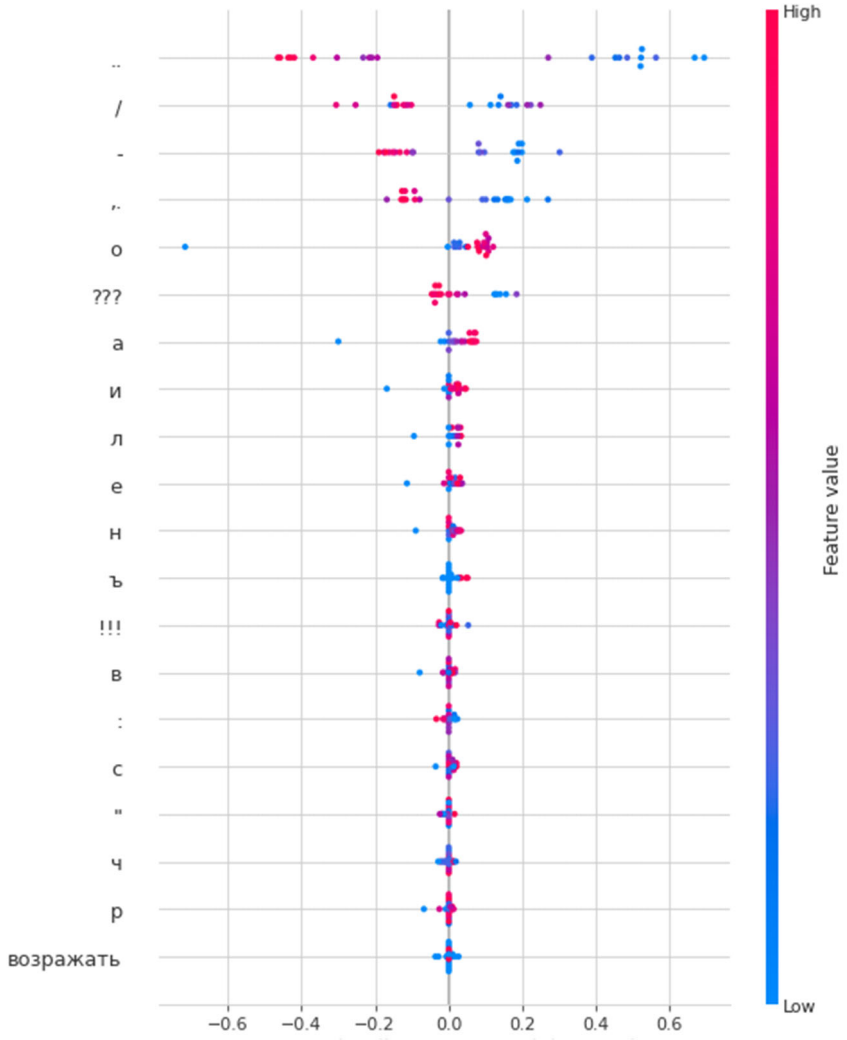


Рисунок 3.6 – Информативные признаки SHAP

Конкретные примеры подтверждают общие результаты (см. таблицу 3.2) и выводы о неэффективности методов отбора с полным перебором (FS и BS). Их результативность не превосходит результаты RS и ГА ни в одном из случаев небинарной классификации.

Признаки, выделенные как информативные методом SHAP, в сочетании с SVM показывают конкурентоспособные результаты только при небольшом количестве признаков и малом числе классов (2, 5, 10), так как особенность метода – эффективность для низкой размерности признакового пространства.

Метод RS позволил добиться лучшей точности для всех рассмотренных случаев. Решающую роль в данном случае сыграл выбор SVM с линейным ядром, которое сочетается с основным ограничением RS – применимостью исключительно к линейным моделям.

Для наиболее эффективных алгоритмов отбора информативных признаков была проведена оценка статистической значимости результатов. В рамках тестов оценивались результаты, полученные SVM в процессе перекрестной проверки без отбора признаков, с отбором RS и с отбором ГА для 2, 5, 10, 20 и 50 авторов. Оценка выполнялась с помощью тестов Фридмана и Неменьи. Согласно полученным результатам, *p*-значение в среднем составило 0,0018 для 2 авторов, 0,0015 – для 5, 0,0003 – для 10, 0,0005 – для 20 и 0,0002 для 50. Исходя из того, что ни в одном из случаев *p*-значение не превысило 0,05, полученная разница считается существенной. Графически статистическую значимость можно изобразить диаграммой Демшара [63] (рисунок 3.7). Она демонстрирует разницу средних рангов пар моделей. Если такая разница меньше критической (CD), то она незначительна.

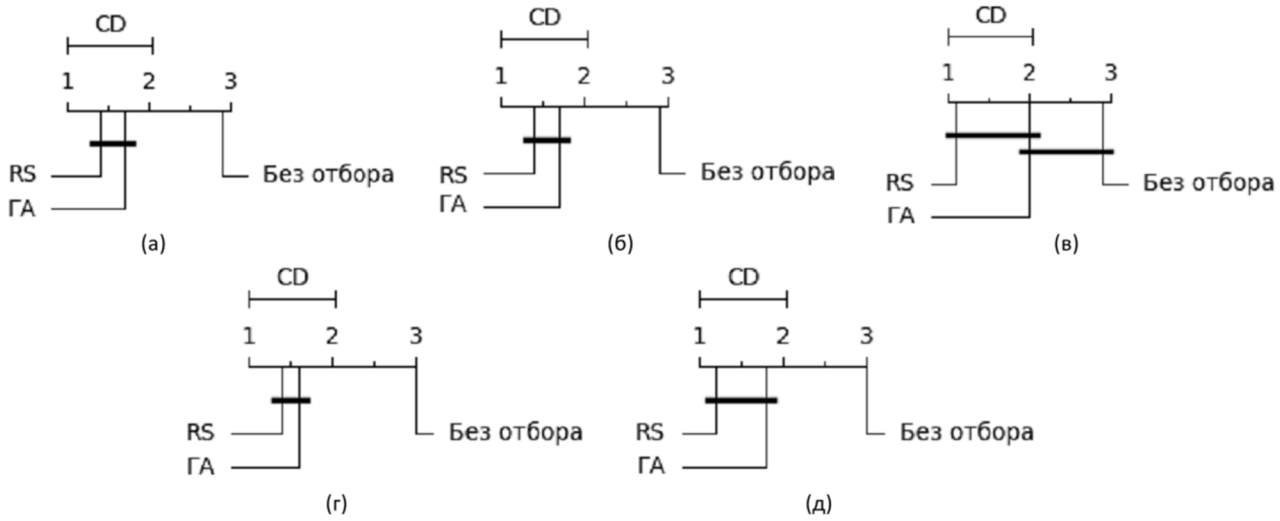


Рисунок 3.7 – Диаграмма Демшара для 2 (а), 5 (б), 10 (в), 20 (г) и 50 авторов (д)

Исходя из полученных результатов, RS отбор позволяет добиться лучших результатов, однако статистическая разница между ГА и RS значительной не является. Подход к обучению SVM без отбора признаков является наименее эффективным и его разница в сравнении с методами отбора ГА и RS статистически значима.

3.3.1.3 Нейронные сети

В таблице 3.3 приведены точности для корпусов из 2, 5, 10, 20 и 50 авторов НС LSTM, BiLSTM, CNN, LSTM+CNN, CNN+LSTM, CNN+CNN, GRU, BiGRU, CNN+GRU, GRU+CNN, BiGRU+CNN, CNN+BiGRU, fastText, RuBERT, Multi-BERT, а также время обучения для самого сложного случая – классификации 50 авторов.

Таблица 3.3 – Результаты определения автора художественного текста нейросетевыми методами

Метод	Количество авторов					Время обучения, 50 авторов, с.
	2	5	10	20	50	
	Точность, %					
LSTM	94,3±5,5	86,7±6,3	75,4±3,3	63,9±5,7	44,2±6,1	10133
BiLSTM	95,5±4,5	82,5±4,8	70,2±5,3	58,7±4,9	41,1±5,1	12284
CNN	97,1±3,6	95,9±2,8	81,3±5,9	71,2±5,8	50,1±5,1	7319
CNN+LSTM	94,5±6,0	86,9±4,8	78,2±5,0	65,8±3,2	50,3±5,2	9264
LSTM+CNN	98,5±5,6	95,5±3,9	82,2±5,4	62,2±5,8	41,3±4.5	9452
CNN+CNN	96,2±4,1	93,7±3,4	86,5±6,1	72,8±4,4	44,8±4,3	8267
GRU	92,1±3,0	90,7±5,0	70,9±3,6	60,8±2,7	40,9±3,6	8381
BiGRU	85,1±3,6	78,2±3,9	68,3 ±2,7	58,4±3,8	38,5±4,2	8665
CNN+GRU	89,0±5,2	90,5±3,7	77,9±2,5	67,5±3,4	45,0±2,9	7845
GRU+CNN	97,6±1,4	95,9±5,0	88,5±4,2	70,1±4,0	56,9±4,2	8028
BiGRU+CNN	85,3±2,9	82,9±4,6	66,9±5,0	58,8±4,5	41,5±3,8	8071
CNN+BiGRU	95,4±5,8	87,6±2,7	75,2±3,4	62,7±4,1	46,5±3,7	7994
fastText	98,2±4,5	95,0±3,7	90,2±6,3	69,9±4,3	45,8±6,2	2372
RuBERT	87,4±5,1	74,8±3,3	69,9±4,1	58,1±2,2	36,9±2,9	19114
MultiBERT	78,7±2,5	73,6±3,7	64,4±4,0	52,2±3,7	27,8±3,4	21095
SVM+RS	98,6±1,4	98,1±1,5	94,2±1,8	88,3±2,5	56,9±2,2	34264

Лучшие результаты были продемонстрированы моделями SVM с RS отбором, GRU+CNN, fastText и CNN. Для перечисленных моделей были проведены тесты, направленные на оценку статистической значимости результатов (рисунок 3.8).

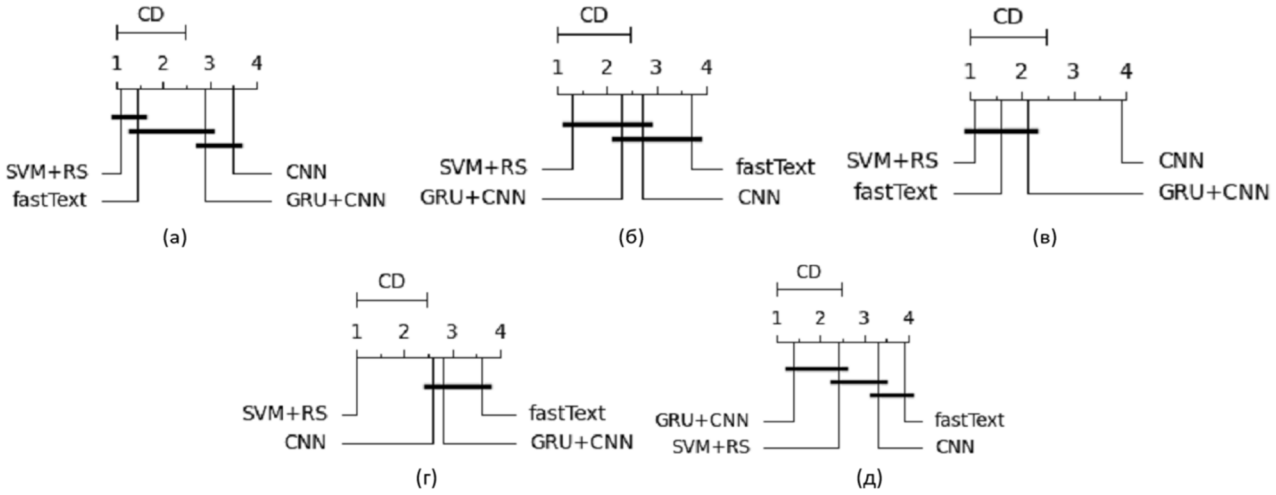


Рисунок 3.8 – Диаграмма Демшара для 2 (а), 5 (б), 10 (в), 20 (г) и 50 (д) авторов

Для всех рассмотренных случаев *p*-значение в рамках теста Фридмана не превысило 0,05, следовательно, полученная разница является существенной. Согласно диаграмме Демшара, для 2, 5, 10 и 20 авторов SVM+RS показывают лучший результат. Для 50 авторов статистическая разница между SVM+RS и GRU+CNN является незначительной, однако лучший результат демонстрирует комбинация из глубоких НС. Отметим также, что метод RS требует в среднем 3 раза больше времени на отбор признаков, чем обучение моделей НС.

3.3.1.4 Оценка точности методики в зависимости от количества и длины образцов

Для оценки разработанной методологии на данных различных объемов было проведено исследование, используя в качестве материалов творчество пяти случайных авторов художественной литературы, с применением кроссвалидации. В ходе экспериментов анализировались от 3 до 20 текстов каждого автора, причем длина текстов варьировалась от 5000 до 20000 символов с интервалом в 1000 символов. Один образец такого же размера используется для тестирования. Детальная информация, включающая результаты каждого отдельного эксперимента, представлена в таблице 3.4.

Таблица 3.4 – Оценка точности методики в зависимости от количества и длины образцов

Объем текста, симв.	Количество текстов автора																	
	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
5000	20±3	25±3	32±3	48±8	55±6	63±4	69±4	72±6	70±3	72±5	74±6	85±4	93±3	93±2	89±2	94±3	96±2	95±2
6000	23±6	25±5	38±5	43±7	57±5	64±6	70±5	78±5	80±3	78±6	80±7	82±6	91±5	86±5	92±5	88±3	94±3	95±3
7000	27±7	29±6	42±6	51±3	62±6	70±7	74±6	84±7	87±4	89±7	85±8	88±5	96±4	94±4	94±6	93±4	95±4	93±4
8000	33±6	36±5	45±5	60±4	65±5	74±6	77±5	85±7	89±4	95±3	94±6	91±5	89±7	97±2	98±2	99±2	98±2	95±4
9000	37±5	45±7	53±8	56±5	66±6	73±5	75±3	89±8	93±5	94±6	97±3	93±6	92±6	95±3	98±1	98±1	98±1	98±4
10000	44±4	50±3	55±7	56±3	60±7	69±8	70±4	84±7	93±6	85±5	92±4	95±5	93±5	96±3	94±6	98±2	95±6	91±5
11000	50±8	55±6	64±6	63±4	67±6	75±7	75±5	82±7	91±7	89±7	85±7	96±7	97±3	98±2	98±3	96±3	97±5	99±2
12000	55±6	58±7	65±7	67±4	72±5	78±3	84±6	91±6	91±4	92±6	93±5	93±6	94±4	97±3	96±2	94±4	95±4	95±6
13000	66±7	67±6	71±6	72±5	72±6	76±4	89±6	93±7	90±4	93±8	88±4	96±4	92±4	93±5	89±3	99±3	99±3	95±4
14000	81±4	82±3	78±2	83±3	85±5	89±1	93±3	98±3	92±3	92±6	93±4	99±1	97±3	94±4	95±4	95±3	95±3	99±2
15000	92±2	92±3	93±2	95±2	94±3	95±2	97±1	99±1	98±1	97±5	95±3	95±4	99±1	97±4	96±3	98±2	98±2	99±1
16000	91±3	89±4	87±3	90±3	94±1	92±3	97±1	96±2	98±4	96±2	95±4	93±4	99±1	99±2	97±2	96±3	99±3	99±2
17000	90±8	92±5	86±4	84±5	87±3	91±3	94±4	93±4	95±6	94±4	93±5	94±1	96±5	98±3	96±3	97±2	95±4	98±2
18000	90±5	90±6	93±2	91±4	88±4	90±4	89±2	90±3	98±2	91±4	88±4	94±5	99±1	96±4	99±2	99±2	99±3	97±2
19000	90±4	87±7	92±4	86±5	79±3	88±6	90±3	92±1	91±5	94±1	95±2	98±2	98±1	97±1	95±3	99±3	99±2	96±3
20000	91±7	93±6	93±3	96±2	92±5	84±5	91±8	94±4	90±8	92±5	93±5	95±5	93±2	94±3	96±3	93±5	98±4	94±3

Согласно результатам экспериментов, установлена длина текстового образца: 15000 символов, количество образцов для обучения: 3 (всего 45000 символов). Точность классификации в этом случае составляет более 90%. При меньшем объеме одного образца сопоставимая точность была получена при использовании 15 текстов по 5000 символов (всего 75000 символов).

3.3.2 Тексты авторов-любителей

Рассматривается два случая:

1) кросс-тематический – тексты созданы на основе разных произведений. В данном случае предложенная ранее методика верифицируется на другом корпусе текстов, написанных не профессиональными писателями. Задача облегчается за счет разной семантики текстов. Тексты относятся к разным категориям, каждая из которых содержит свои уникальные слова и речевые обороты.

2) монотематический – тексты созданы на основе одного произведения. Для решения практических задач это можно интерпретировать как подражание оригинальному авторскому стилю. Задача усложняется, потому что в текстах одной категории используются одни и те же ключевые слова и словосочетания. В результате тексты становятся похожими по внутреннему содержанию, а для поиска отличительных признаков следует проводить глубокий анализ семантики и опираться исключительно на особенности стиля письма конкретного автора.

Постановка эксперимента в части длины и количества образцов аналогична определению авторства художественного текста. Использовались модели, демонстрирующие лучшие результаты идентификации автора художественного текста.

3.3.2.1 Определение авторства в кросс-тематическом случае

Результаты идентификации автора для корпуса любительских текстов приведены в таблице 3.5.

Таблица 3.5 – Результаты определения автора любительского текста в кросс-тематическом случае

Модель	Количество авторов					Время обучения, 50 авторов, с.
	2	5	10	20	50	
	Точность, % (кол-во подобранных признаков)					
fastText	99,0±1,9	95,2±4,0	91,0±4,1	74,8±3,4	57,5±5,0	2372
GRU+CNN	96,6±2,9	95,7±6,9	92,6±5,8	74,2±4,8	61,8±6,1	8028
BiGRU+CNN	98,9±1,7	92,3±4,8	91,8±3,0	65,8±4,7	50,6±4,2	8071
CNN+BiGRU	96,1±2,8	95,0±2,1	83,3±3,8	69,1±3,5	56,9±6,1	7994
SVM	89,9±3,4	86,9±4,6	83,0±3,3	69,5±3,2	52,1±5,5	118
SVM+RS	98,9±1,2 (1994)	96,3±1,8 (2841)	90,1±2,1 (3402)	74,7±3,3 (4773)	61,6±2,8 (5281)	73904
SVM+ГА	98,6±0,4 (2367)	96,7±1,3 (2745)	91,9±1,8 (2779)	73,1±2,4 (4987)	62,4±3,5 (5332)	69875

Среди нейросетевых подходов можно отметить fastText и GRU+CNN. Классические методы с отбором не уступают нейросетевым моделям, однако подбор параметров с их помощью занимает на порядок больше времени, чем обучение глубоких моделей.

В данном эксперименте не выявлено модели, демонстрирующей лучший результат для всех рассматриваемых наборов авторов. Для выбора оптимального варианта из всех альтернатив рассмотрены критерии принятия решений в условиях риска [332]. В качестве возможных состояний среды, в которых могут быть использованы альтернативы – 2, 5, 10, 20 и 50 авторов. Вероятность наступления каждого состояния задана одинаково (P=0,2).

Исходя из рассчитанных значений критериев, можно сделать вывод, что GRU+CNN является оптимальным выбором (таблица 3.6):

1) по критерию Байеса – Лапласа выбрана модель GRU+CNN, так как имеет максимальное байесовское значение функции полезности;

2) согласно максиминному критерию Вальда, выбор происходит на основе сравнения минимальных значений каждой альтернативы и выбора максимальной из них. Выбор в пользу GRU+CNN;

3) при рассмотрении критерия минимаксного риска Сэвиджа формируется матрица рисков, где для каждой альтернативы выбирается максимальное для состояний значение, а затем выбирается альтернатива с минимальным значением из этих максимальных, то есть с минимальной «упущенной выгодой». Выбор в пользу GRU+CNN.

Таблица 3.6 – Расчетные значения критериев принятия решений в условиях риска

Метод	Байеса – Лапласа	Максиминный	Минимаксный
fastText	82,1	57,5	4,3
GRU+CNN	**82,8**	**61,8**	**2,4**
CNN+BiGRU	78,1	50,6	11,2
BiGRU+CNN	78,7	56,9	9,2
SVM	74,9	52,1	9,7
SVM+RS	74,8	56,6	16,7
SVM+ГА	74,7	55,5	14,3

3.3.2.2 Определение автора в монотематическом случае

В монотематической атрибуции использовались авторы, создававшие свои тексты на основе серии книг о Гарри Поттере. Результаты приведены в таблице 3.7.

Таблица 3.7 – Результаты определения автора любительского текста в монотематическом случае

Модель	Количество авторов					Время обучения, 50 авторов, с.
	2	5	10	20	50	
	Точность, % (кол-во подобранных признаков)					
fastText	96,6±1,6	91,5±2,0	75,9±3,5	64,6±2,4	46,7±2,9	2280
GRU+CNN	96,8±1,8	93,9±3,1	78,4±3,9	68,2±3,0	51,3±3,4	8004
BiGRU+CNN	90,0±4,5	79,4±5,2	66,6±3,6	55,3±4,2	35,0±4,2	8142
CNN+BiGRU	97,9±3,0	86,6±2,9	75,8±2,0	66,9±2,9	45,9±4,3	8012
SVM	93,6±2,2	81,1±1,8	72,7±2,9	62,6±1,4	39,8±4,0	109
SVM+RS	93,1±1,2 (2144)	90,5±0,9 (2366)	77,6±1,7 (3057)	61,5±3,4 (4648)	46,4±2,1 (5753)	62288
SVM+ГА	93,2±0,9 (2590)	90,0±1,6 (2557)	83,5±1,9 (2836)	67,3±2,4 (5348)	48,1±3,3 (6567)	60136

При той же постановке эксперимента, результаты определения автора в кросс-тематическом случае на 2–14% выше, чем в случае монотематической атрибуции, ввиду сложности последней. Все модели демонстрируют точность 90% и более при классификации двух авторов. Лучшие результаты во всех случаях, кроме бинарного, продемонстрировала модель GRU+CNN. Это подтверждает правильность ее выбора, как оптимальной альтернативы по критериям принятия решений в условиях риска.

3.3.3 Короткие тексты пользователей социальных сетей

Учитывая специфику коротких текстов, было принято решение провести полный цикл экспериментов, включая классические методы МО и НС и отбор информативных признаков, аналогично художественным текста.

3.3.3.1 Классические методы

Результаты исследований классических методов машинного обучения для коротких сообщений пользователей социальных сетей представлены таблице 3.8.

Таблица 3.8 – Результаты определения автора короткого текста

Метод	Количество авторов					Время обучения, 50 авторов, с.
	2	5	10	20	50	
	Точность, %					
Обучение на признаковом пространстве						
SVM	82,2±4,0	69,9±3,5	66,3±3,8	55,3±3,1	32,1±3,9	79
LR	77,1±3,1	69,5±4,2	48,2±2,9	34,3±3,4	28,6±3,6	73
NB	72,9±2,3	68,6±2,6	45,9±3,5	38,8±4,1	27,1±3,3	59
DT	79,1±2,1	63,5±2,1	24,2±3,6	19,9±4,1	15,9±3,3	57
RF	81,2±3,9	56,1±2,8	37,6±2,7	32,2±1,7	25,9±2,4	57
KNN	78,1±4,2	65,4±3,7	61,9±4,0	43,9±4,1	33,6±3,4	64
Обучение на TF-IDF представлении текста						
SVM	61,1±3,1	54,4±6,0	39,4±3,9	32,0±1,0	17,3±2,6	68
LR	69,4±5,2	58,1±4,4	44,7±0,7	36,1±2,2	24,8±3,1	60
NB	70,8±1,0	66,1±7,3	49,6±5,4	46,1±2,3	34,1±5,0	49
DT	69,1±2,1	43,5±2,1	24,2±3,6	15,4±3,6	11,0±4,8	45
RF	68,4±2,9	60,4±3,2	42,6±1,9	33,6±2,2	23,5±1,9	51
KNN	57,9±1,7	54,2±3,1	45,3±1,9	40,1±2,3	32,7±2,45	58

В отличии от классификации художественных текстов, полученные результаты позволяют сделать вывод о неэффективности классических методов МО при использовании сформированного признакового пространства. Это связано с тем, что длины комментариев очень малы. Содержимое комментариев отражает эмоции автора по отношению к комментируемому событию, поэтому в корпусе преобладают односложные высказывания и короткие предложения. SVM, обученный на экспериментально подобранных параметрах и признаковом пространстве, достигает максимальной точности 82% для двух авторов. Использование TF-IDF вместо вектора признаков дало преимущество только для двух моделей, LR и NB, в экспериментах с 10 и более авторами.

3.3.3.2 Отбор признаков

Для коротких комментариев был проведен первичный эксперимент с использованием ГА, в результате которого установлено отсутствие эффективности классических методов МО для текстов такого типа (таблица 3.9). Параметры ГА аналогичны представленным в п. 3.2.3.

Таблица 3.9 – Отбор признаков для коротких текстов (RS, ГА)

Алгоритм отбора	Количество авторов					Время обучения, с.
	2	5	10	20	50	
	Точность, % (кол-во подобранных признаков)					
RS	89,5±1,3 (2776)	88,3±2,0 (2959)	78,0±3,1 (3716)	71,8±2,2 (4219)	47,9±2,6 (6190)	96284
ГА	91,3±0,6 (2568)	88,5±0,3 (2286)	78,2±2,0 (3565)	70,8±2,5 (3168)	50,2±2,0 (6136)	98753
1168 признаков [289]	82,2±4,0	69,9±3,5	66,3±3,8	55,3±3,1	32,1±3,9	79
Без отбора, TF-IDF	61,1±3,1	54,4±6,0	39,4±3,9	32,0±1,0	17,3±2,6	68
Без отбора, 7151 признак	82,0±2,8	78,3±3,3	69,9±0,9	50,7±1,3	33,2±2,0	131

Результаты, представленные в таблице 3.10, свидетельствуют об улучшении точности классификации при применении методов отбора на 7–16%. Однако временные затраты на отбор затрудняют применение подходов при решении реальных практических задач, т.к. для каждого нового случая требуется проводить отбор заново. Использование НС (см. пп. 3.3.3.3) позволяет добиться большей точности, при этом временные затраты на порядок меньше. Поэтому дальнейшие эксперименты с методами отбора для коротких текстов не проводились, а нейросетевой подход выбран как основной.

3.3.3.3 Нейронные сети

Аналогично исследованиям для художественных текстов, в таблице 3.10 представлены результаты исследований нейросетевых методов для коротких сообщений пользователей социальных сетей.

Таблица 3.10 – Результаты определения автора короткого сообщения

Метод	Количество авторов					Время обучения (50 авторов), с.
	2	5	10	20	50	
	Точность, %					
LSTM	93,0±1,9	89,7±1,9	73,0±2,8	68,8±2,4	50,1±2,6	6012
BiLSTM	94,6±2,1	92,5±2,4	71,3±2,4	59,3±1,3	49,6±3,9	8596
CNN	95,6±2,0	93,3±1,5	72,4±2,7	67,9±3,3	48,8±3,6	3499
CNN+LSTM	95,5±3,5	90,9±2,2	77,1±3,9	62,2±3,4	47,3±1,9	4610
LSTM+CNN	92,3±2,1	90,2±1,0	64,2±3,3	61,3±3,2	47,4±2,8	7677
CNN+CNN	91,3±2,4	89,2±2,2	76,6±4,5	69,8±5,8	50,2±1,4	3683
GRU	91,0±5,0	90,1±5,3	70,6±4,4	64,7±7,5	52,0±6,1	4020
BiGRU	92,5±4,2	90,2±4,0	72,3±3,3	63,0±5,4	48,9±6,0	4431
CNN+GRU	90,3±3,8	85,4±3,2	71,1±4,9	68,2±6,7	55,2±5,3	3715
GRU+CNN	96,3±5,5	93,5±6,2	78,1±5,4	73,7±2,5	56,0±5,5	3920
BiGRU+CNN	96,0±3,9	92,0±3,7	73,9±2,9	71,6±7,0	53,1±4,2	4157
CNN+BiGRU	96,0±3,5	93,4±5,9	78,0±4,8	66,7±4,9	47,5±3,8	3822
fastText	94,0±1,2	87,2±2,2	76,1±3,5	68,4±2,3	55,6±2,8	1017
RuBERT	93,3±2,2	88,6±1,8	76,6±3,2	66,8±3,3	50,0±2,9	15472
MultiBERT	90,2±1,9	87,1±2,2	69,5±3,0	63,4±2,7	47,1±2,8	18093
SVM+RS	89,5±1,3 (2776)	88,3±2,0 (2959)	78,0±3,1 (3716)	71,8±2,2 (4219)	47,9±2,6 (6190)	96284
SVM+ГА	91,3±0,6 (2568)	88,5±0,3 (2286)	78,2±2,0 (3565)	70,8±2,5 (3168)	50,2±2,0 (6136)	98753

Для всех рассматриваемых случаев лучший результат был достигнут сочетанием архитектур GRU+CNN, ранее продемонстрировавшим высокую точность при определении автора художественного текста для корпуса, включающего 50 авторов. Точность такой комбинации достигает 96,3%. Сопоставимые метрики качества демонстрируют модели BiGRU+CNN, CNN+BiGRU и fastText. Исходя из этого, возникает необходимость в оценке статистической значимости полученных

результатов тестами Фридмана и Неменьи. В рамках данных тестов *p*-значение составило менее 0,05, значит полученная разница является существенной. Согласно диаграмме Демшара (рисунок 3.9), лучшие результаты демонстрирует GRU+CNN, она же имеет первый ранг по всем проведенным измерениям, однако статистическая разница с BiGRU+CNN/CNN+BiGRU не является значимой ни в одном из рассмотренных случаев.

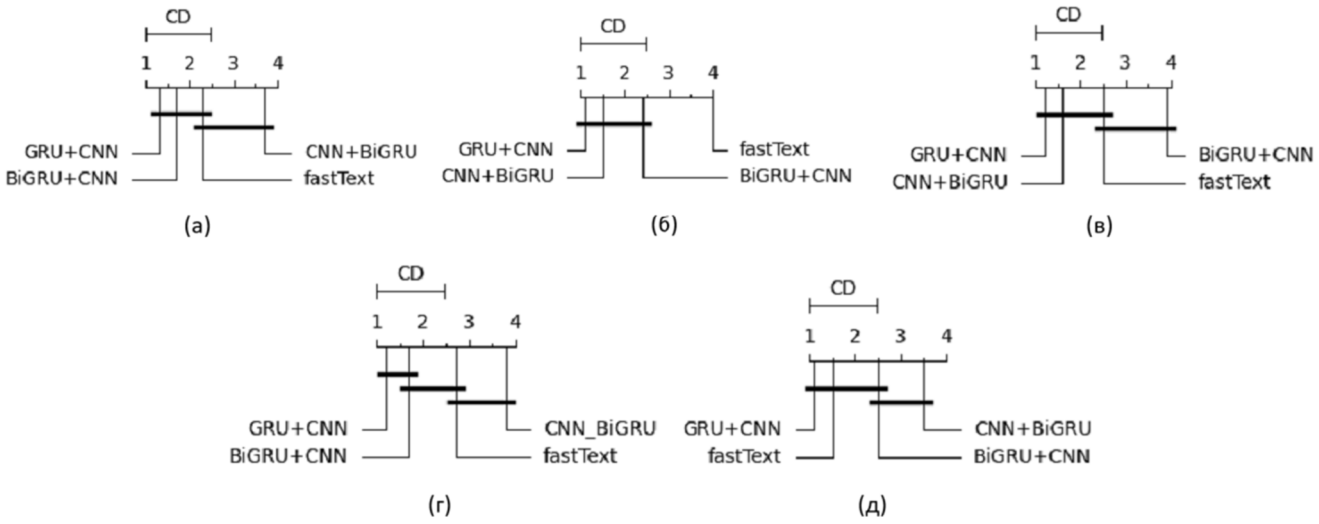


Рисунок 3.9 – Диаграмма Демшара для 2 (а), 5 (б), 10 (в), 20 (г) и 50 (д) авторов

3.3.4 Случай использования генеративных моделей

Для оценки методики в случае анализа текстов, созданных новейшими генеративными моделями, было проведено два эксперимента. Первый (таблица 3.11) посвящен классификации искусственных текстов, созданных на основе естественных текстов (коротких комментариев) авторов. В эксперименте определяется, кто из авторов-«клонов», является автором сгенерированного сообщения. В качестве метода классификации использовался GRU+CNN.

Из результатов видно, что с увеличением количества авторов точность классификации систематически снижается для обоих корпусов. Так, при переходе от 2 до 50 авторов точность классификации уменьшается с 93,2% до 50,7%, что подчеркивает сложность классификации при большем количестве авторов.

Таблица 3.11 – Классификация сгенерированных комментариев

Количество авторов	Точность на GPT-3.5, %	Точность на GPT-4, %
2	93,2±4,4	95,3±7,2
5	81,7±2,9	84,2±6,1
10	75,3±4,7	77,4±4,9
20	63,4±4,6	66,6±3,8
50	50,7±6,1	52,3±4,4

При работе с данными, созданными GPT-4, GRU+CNN демонстрирует более высокую точность классификации по сравнению с данными GPT-3.5 во всех экспериментах. Однако более высокие значения стандартных отклонений у GPT-4 также свидетельствуют о возможной чувствительности модели к особенностям датасета и вариативности стилей письма.

Второй эксперимент проводился на смешанном наборе данных, включающем оригинальные тексты авторов, а также сгенерированные на их основе образцы авторов-моделей. Результаты приведены в таблице 3.12.

Таблица 3.12 – Классификация на смешанном наборе данных

Количество авторов (классов)	Точность SVM+RS, %		Точность GRU+CNN, %	
	Оригинальные тексты + GPT-3.5	Оригинальные тексты+GPT-4	Оригинальные тексты + GPT-3.5	Оригинальные тексты + GPT-4
1 (2)	92,6±4,1	85,2±4,0	93,5±3,2	91,3±4,1
2 (4)	80,1±3,9	72,5±3,4	90,0±1,2	89,9±3,3
5 (10)	65,9±3,5	63,1±5,1	84,2±2,2	82,1±5,0
10 (20)	60,3±3,8	54,4±4,2	64,1±3,5	60,7±3,6
20 (40)	48,3±3,1	41,6±5,0	54,4±2,3	47,8±3,2
50 (100)	30,1±3,9	23,2±6,2	39,6±2,8	34,3±4,1

Во всех случаях GRU+CNN показывает лучшие результаты классификации. Оригинальные тексты автора можно отличить от образцов, сгенерированных GPT, с точностью 91,3% и 93,5% для версий 4 и 3.5 соответственно. Максимальная потеря точности для GRU+CNN составила 19%, а для SVM – 17%. Наибольшая потеря точности зафиксирована при проведении экспериментов для 50 и 20 авторов, в то время как для 2 и 5 авторов потери не превышают 8% от результата, полученного без использования сгенерированных образцов.

3.4 Методика верификации автора

Для реализации механизма верификации авторства предлагается использовать One-Class SVM, обученный на текстах автора (рисунок 3.10).

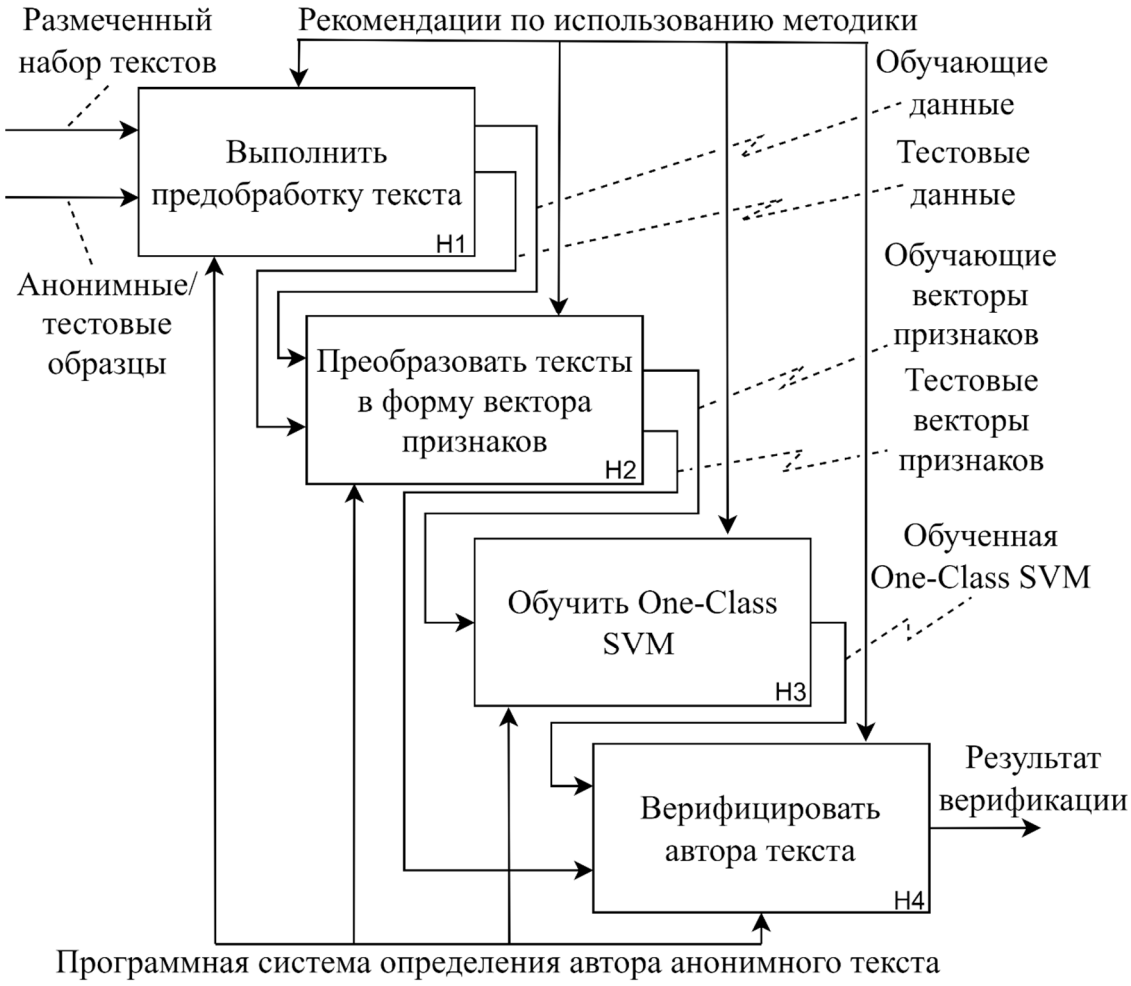


Рисунок 3.10 – Методика верификации авторства текста

При апробации метода была проведена серия экспериментов на художественных, любительских и коротких текстах. Тексты были преобразованы в форму вектора для информативных признаков или в TF-IDF представление.

Обучение одноклассовых классификаторов происходило на текстах каждого из авторов: 3 для длинных текстов, 40 для коротких.

На вход каждой обученной модели подавались тексты как самого автора, так и образцы других авторов. Так, к тестовой выборке были добавлены образцы 1, 4,

9, 19, 49, 99 авторов. Для тестирования использовался 1 текст каждого автора для длинных и 10 для коротких текстов.

Итоговые параметры One-Class SVM, подобранные жадным поиском: линейное ядро, коэффициент регуляризации – 0,1, допуск по критерию останова – 0,00001.

Результаты исследования приведены в таблице 3.13.

Таблица 3.13 – Результаты экспериментов по верификации автора текста

Количество авторов при тестировании	Представление текста	Полнота, %		
		Художественные	Любительские	Короткие
2	TF-IDF	98,9±0,5	98,3±2,2	96,5±1,6
	Признаки	99,2±0,3	98,9±1,5	97,8±1,1
5	TF-IDF	97,4±2,0	96,4±3,2	96,4±0,6
	Признаки	96,1±0,9	98,7±1,1	97,4±2,0
10	TF-IDF	96,9±2,2	93,9±2,5	95,7±2,1
	Признаки	95,2±1,3	97,4±1,9	96,3±3,0
20	TF-IDF	87,2±4,0	87,0±2,9	95,3±3,8
	Признаки	93,1±3,1	90,2±4,1	96,2±4,9
50	TF-IDF	79,2±3,9	78,5±2,1	95,3±5,0
	Признаки	86,8±4,4	90,7±4,1	95,6±7,0
100	TF-IDF	69,0±7,7	67,2±8,3	83,4±8,1
	Признаки	80,1±6,5	82,7±6,1	88,2±7,0

При тестировании на текстах 2 авторов метод One-Class SVM показал высокую полноту для всех типов текстов, что свидетельствует о высокой способности метода отличать авторские стили разных авторов. С увеличением количества авторов в тестовой выборке до 100 точность значительно снижается. В большинстве случаев использование информативных признаков давало более высокие результаты по сравнению с TF-IDF представлением.

Полнота верификации автора короткого текста достигает 97,8%, что позволяет сделать выводы о том, что разработанный подход может применяться для продленной аутентификации пользователя на основе создаваемых им текстов.

3.5 Методика идентификации автора для случая открытой атрибуции

Методика открытой атрибуции, представленная в виде IDEF0 диаграммы, изображена на рисунке 3.11.

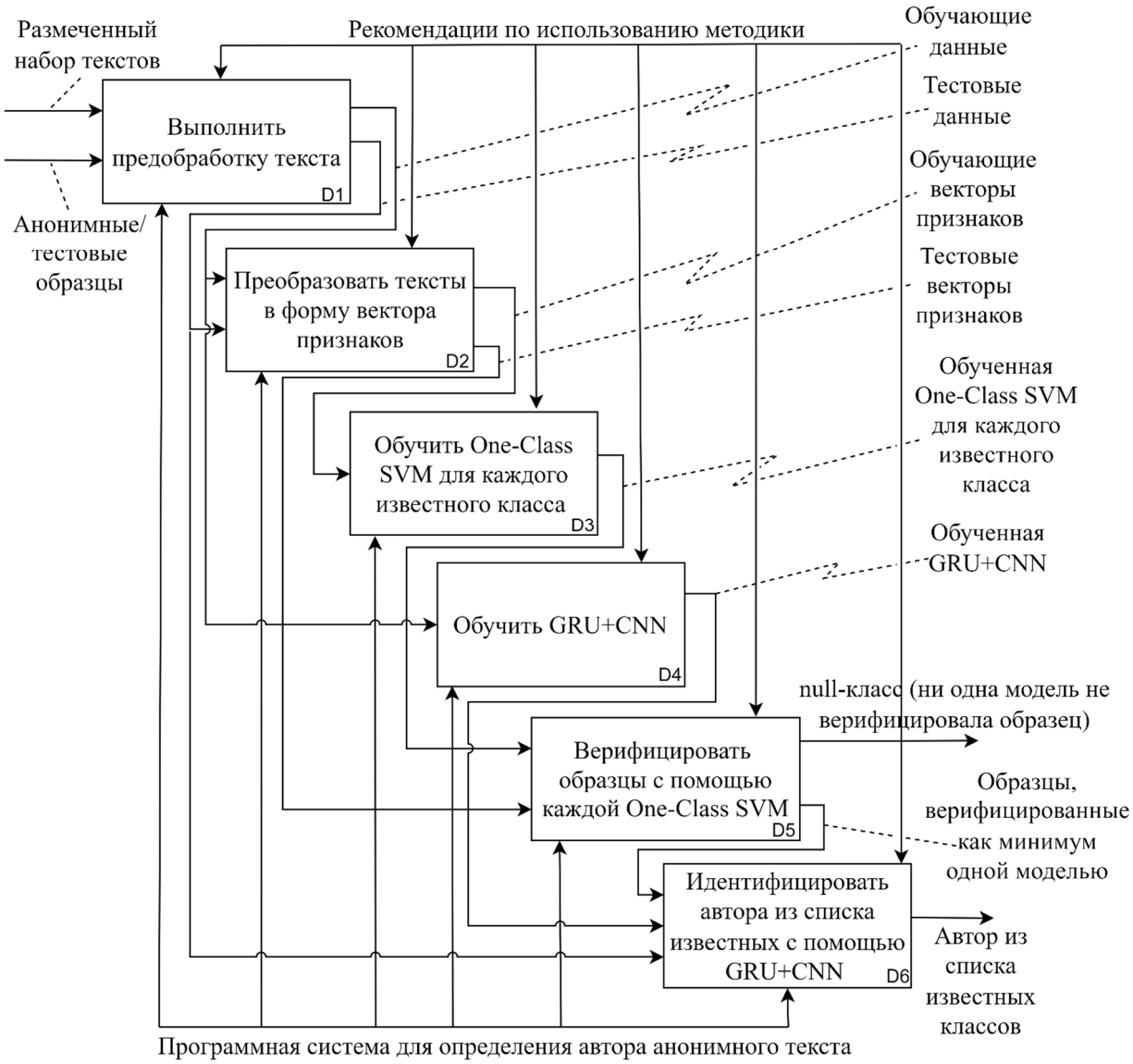


Рисунок 3.11 – Методика определения автора текста (открытая атрибуция)

На начало 2025 года не существует общепринятых методов открытой атрибуции. Поэтому за основу выбрана комбинация:

1) одноклассового SVM с информативными признаками для верификации авторства;

2) GRU+CNN, показавшая лучшие результаты для закрытого случая.

3.5.1 Метод на основе комбинации одноклассового SVM и GRU+CNN

Данный метод включает два этапа.

1. Верификация авторства. Обучение отдельных One-Class SVM на текстах каждого известного автора. Совокупность обученных One-Class SVM используется как инструмент для верификации авторства и детектирования аномалий. При тестировании неизвестного образца, он подается на вход каждого из классификаторов. Если все одноклассовые классификаторы отнесут его к отрицательному классу, то принимается итоговое решение, что текст относится к null-классу (автором текста является неизвестный автор, образцы которого не участвовали в обучении).

2. По итогам работы всех One-Class SVM, все верифицированные образцы подаются на вход классификатора GRU+CNN, обученного только на текстах известных классов. Эта модель идентифицирует конкретного автора.

При оценке точности методики, учитываются как тексты известных классов, так и неизвестных авторов.

3.5.2 Метод на основе распределения вероятностей Softmax при использовании GRU+CNN

Метод основывается на вычислении пороговых значений, ниже которых модель будет считать, что она недостаточно уверена в своем предсказании. GRU+CNN делает предсказания распределения вероятностей для образцов обучающей выборки каждого класса путем применения функции активации SoftMax.

В результате обучения модели на m образцах n классов и f кросс-валидационных фолдах получается матрица распределения вероятностей принадлежности образца одному из классов:

$$\mathbf{P} = \left[p_{111}, p_{ijk}, \ldots, p_{nmf} \right], \tag{3.5}$$

где $i = \overline{1,n}$ – индекс класса; $j = \overline{1,m}$ – индекс образца, $k = \overline{1,f}$ – индекс фолда кросс-валидации.

Для расчета пороговых значений для каждого класса берутся минимальные вероятности принадлежности образца классу в рамках каждого фолда. Далее среди них находятся максимальные значения по всем фолдам:

$$\Theta = [\theta_1, .., \theta_n], \tag{3.6}$$

$$\theta_i = \max_{k=1}^{f} \left(\left[\min_{i=1}^{n} (\mathbf{P}) \right] \right). \tag{3.7}$$

При подаче на вход анонимного образца также получается распределение вероятностей принадлежности образца к каждому из классов $\mathbf{P}' = [p'_1, p'_2, ..., p'_n]$, при этом сумма вероятностей равна 1. В случае, если ни один элемент данного вектора не превышает пороги, данный текст будет классифицирован как выброс (null класс). Если порог превышен только для одного класса, образец относится к этому классу. Если порог превышен для нескольких классов – то к классу, вероятность принадлежности которому максимальна:

$$\begin{aligned} &\forall i, p'_i < \theta_i \Rightarrow t \in \mathbf{T}_{a_{\text{null}}}, \\ &\exists i, p'_i \geq \theta_i \Rightarrow t \in \mathbf{T}_{a_i}, \\ &\exists i, j, i \neq j, p'_i \geq \theta_i, p'_j \geq \theta_j, p'_i > p'_j \Rightarrow t \in \mathbf{T}_{a_i}. \end{aligned} \tag{3.8}$$

3.5.3 Метод на основе порогового значения расстояния Евклида

Метод на основе расстояния Евклида описан в п. 2.7.1

3.5.4 Метод на основе порогового значения косинусного сходства

Последний метод открытой атрибуции основан на пороговом значении косинусного сходства. Подобно предыдущему методу для каждого текста рассчитывается свой вектор признаков, затем для положительных классов рассчитываются

центроиды. Также для каждого обучающего объекта считается мера косинусного сходства по отношению к центроиду класса (п. 2.3.8). Минимальное значение по классу признается пороговым. Тестовые образцы, не превышающие пороговое значение ни для одного класса, относят к отрицательному классу.

3.6 Результаты экспериментов по определению авторства в случае открытого множества кандидатов

При решении задачи идентификации автора в случае открытого множества кандидатов использовались методы, основанные на одноклассовой модификации SVM и GRU+CNN, одиночном использовании GRU+CNN, а также статистические методы, не основанные на МО, поскольку представление текста в виде числового вектора позволяет с ними работать.

Параметры GRU+CNN идентичны описанным в п. 3.2.4. Параметры одноклассового SVM: линейное ядро, коэффициент регуляризации – 0,1, допуск по критерию останова – 0,00001.

Для открытой атрибуции обучение происходило на текстах N авторов, а при тестировании использовались тексты N+M авторов (рисунок 3.12), где

$$M = \lceil N \times 0,3 \rceil = \min\{n \in \mathbb{Z} \mid n \geq (N \times 0,3)\} - \quad (3.9)$$

количество авторов, тексты которых используются как нулевой класс. Для авторов, входящих в N, количество образцов для тестирования и обучения эквивалентно постановке задачи закрытой атрибуции (3 и 1 для художественных и любительских текстов; 40 и 10 для коротких комментариев). Эксперименты проводились для 2+null, 4+null, 9+null, 19+null и 49+null авторов, где второе слагаемое означает нулевой класс, включающий дополнительно 30% текстов для тестирования M авторов от объема тестовой выборки авторов N.

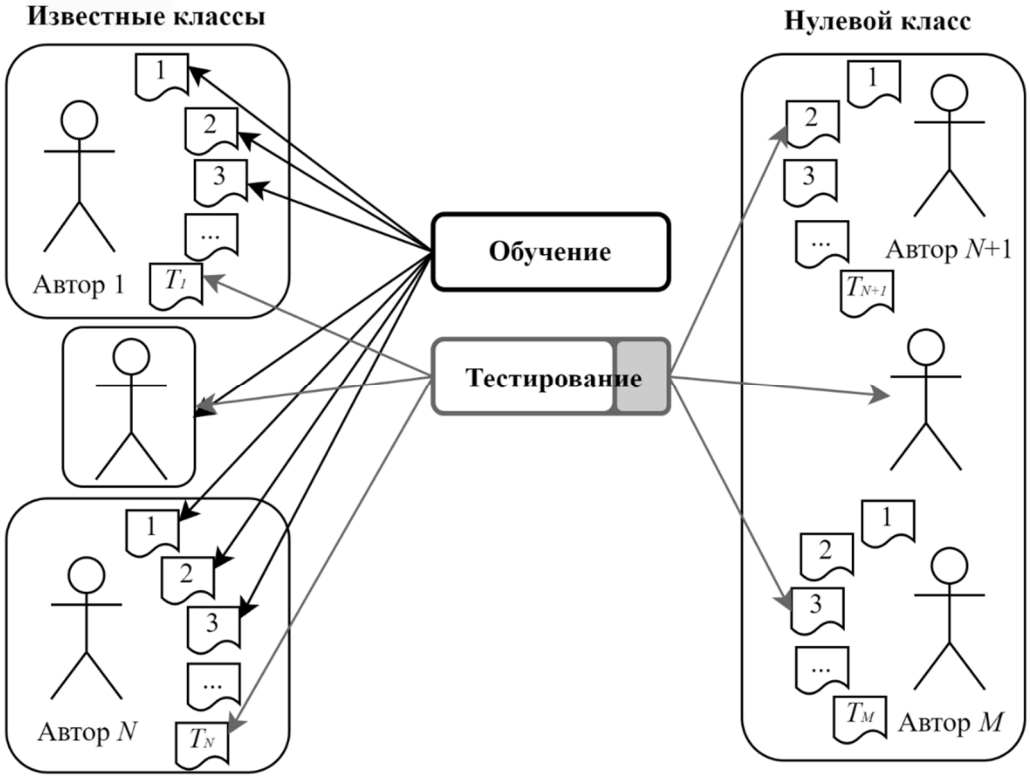


Рисунок 3.12 – Формирование данных для открытого множества кандидатов

3.6.1 Художественные тексты

Результаты статистических методов и подходов, основанных на GRU+CNN, приведены в таблице 3.14.

Таблица 3.14 – Результаты идентификации автора в случае открытого множества кандидатов для художественных текстов

Метод	Количество авторов				
	2+null	4+null	9+null	19+null	49+null
	Точность, %				
Расстояние Евклида	59,9±1,7	55,2±1,6	42,9±3,1	35,4±2,2	26,5±3,0
Косинусное сходство	85,2±1,1	74,4±2,1	45,4±4,5	39,0±2,1	31,3±2,2
GRU+CNN (порог SoftMax)	93,5±3,0	89,0±2,3	82,2±3,4	60,3±4,0	46,3±3,4
One-Class SVM и GRU+CNN	94,2±2,1	89,2±1,9	69,4±2,2	53,8±2,4	40,8±3,4

Согласно результатам, методы, основанные на расчете косинусного сходства и расстояния Евклида, уступают методам МО до 40%. Методы, основанные на GRU+CNN, демонстрируют лучшие результаты во всех проведенных экспериментах. Стоит отметить специфику художественных текстов в контексте открытой атрибуции – соблюдение синтаксических и семантических норм русского языка, богатство лексики, образность и эмоциональность речи в стилях всех авторов-классиков. При создании нулевого класса, используются тексты нескольких разных писателей (кроме случая 2+null авторов), не участвовавших в обучении. За счет этого теряется преимущество, основанное на уникальном стиле профессионального писателя, и повышается сложность классификации. Этим объясняется проигрыш комбинации One-Class SVM и GRU+CNN для 9+null и более авторов. Также отметим, что решение аналогичной выше задачи на практике маловероятно ввиду того, что современный стиль неформального общения контрастирует с художественным.

3.6.2 Тексты авторов-любителей

Эксперименты на любительских текстах проводились для случаев кросстематической (рисунок 3.13) и монотематической (рисунок 3.14) атрибуции.

Метод, основанный на совместном использовании одноклассового SVM и GRU+CNN, позволяет учитывать и сформированное признаковое пространство и проводить семантический анализ текста с помощью НС. Благодаря этому, результаты, полученные с использованием данного метода, превосходят полученные другими методами для всех рассмотренных случаев.

Статистические методы показали худшие результаты в обоих случаях. Разница с подходом на основе SVM и GRU+CNN достигает 15%.

Можно отметить, что разброс результатов в зависимости от источника сообщений и тематической категории находится в пределах 8%, что свидетельствует о хорошей обобщающей способности полученных моделей и применимости разработанной методики к решению реальных задач.

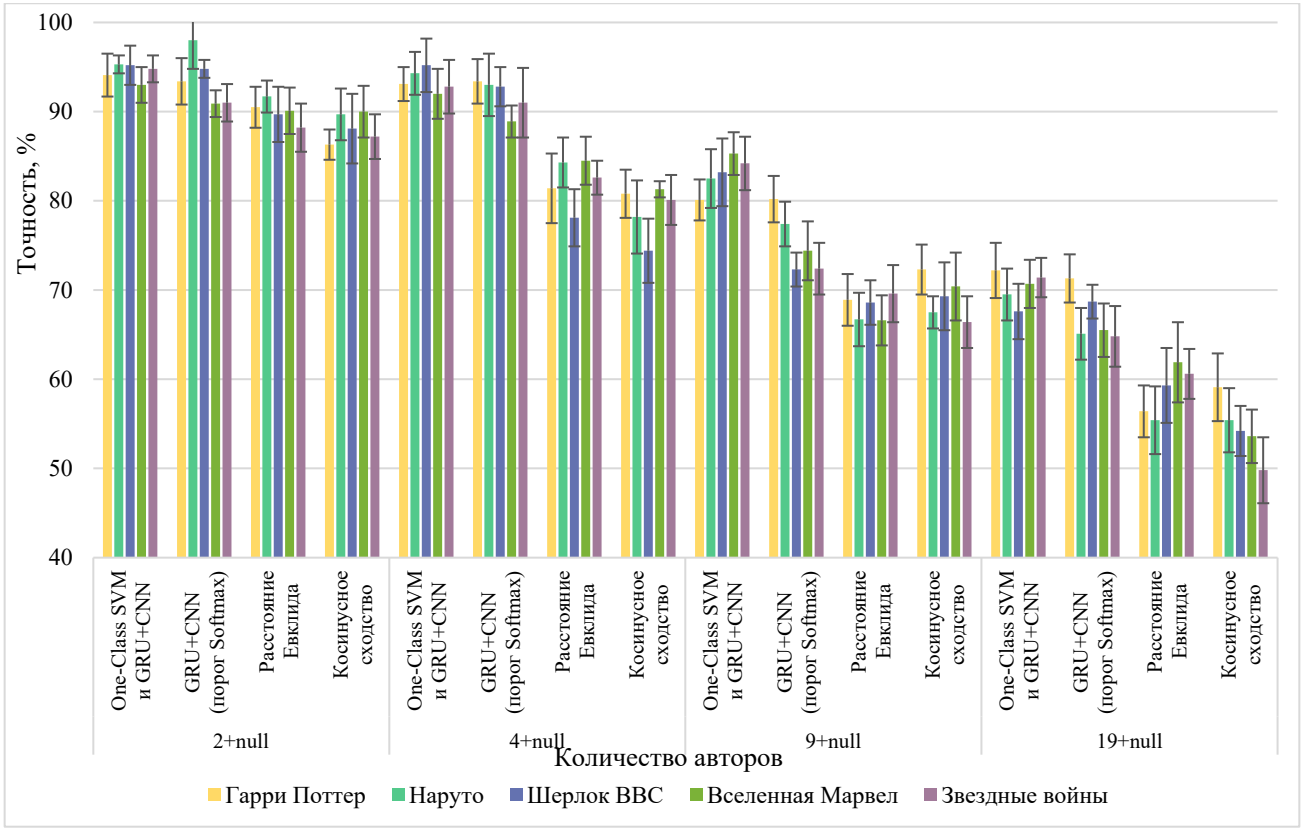


Рисунок 3.13 – Межгрупповая классификация

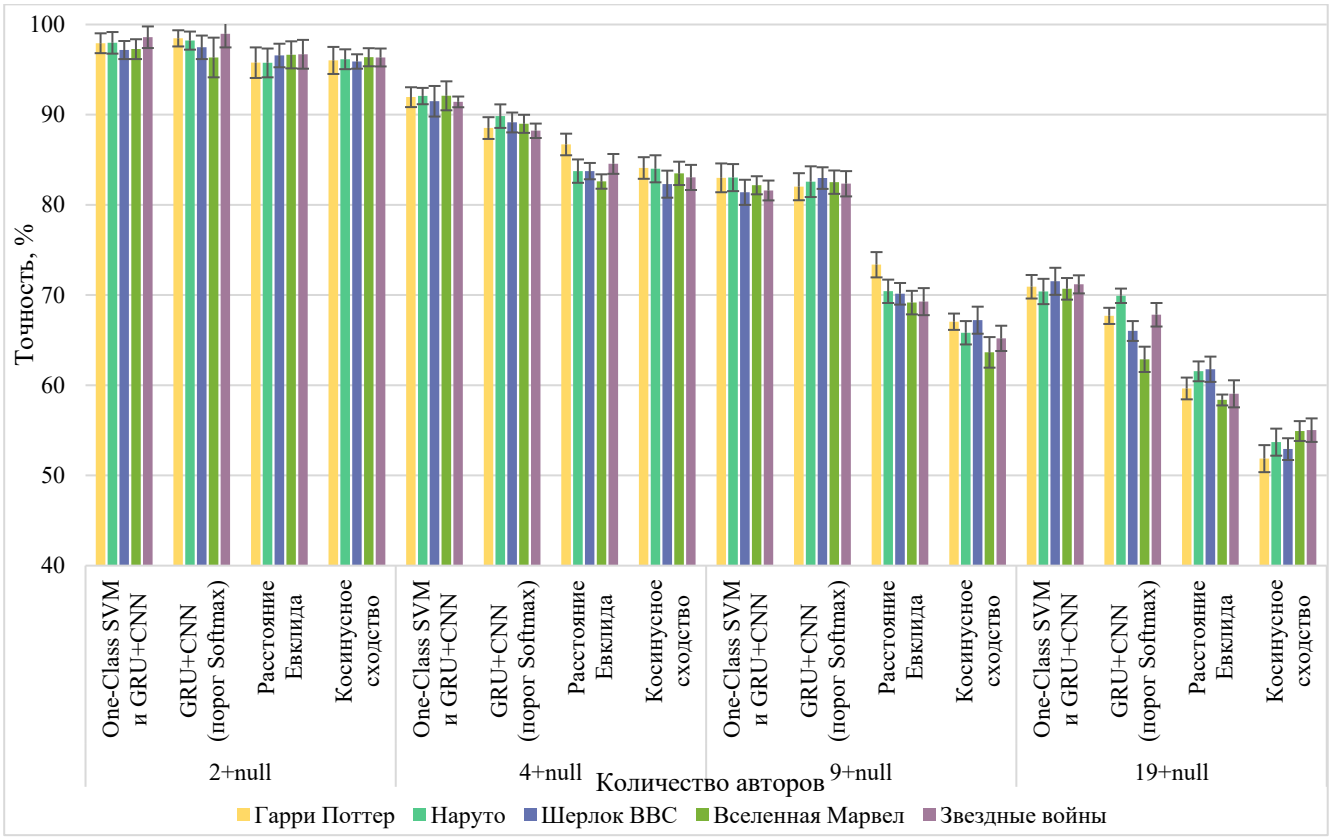


Рисунок 3.14 – Классификация внутри группы

3.6.3 Короткие тексты пользователей социальных сетей

Результаты статистических методов и подходов, основанных на GRU+CNN, приведены в таблице 3.15.

Таблица 3.15 – Результаты идентификации автора в случае открытого множества кандидатов для коротких текстов пользователей социальных сетей

Метод	Количество авторов				
	2+null	4+null	9+null	19+null	49+null
	Точность, %				
Расстояние Евклида	69,8±0,7	57,7±2,1	54,2±1,3	47,4±2,2	25,2±2,8
Косинусное сходство	71,3±2,2	64,6±1,8	48,9±0,9	40,6±2,6	22,4±2,3
GRU+CNN (порог SoftMax)	91,9±2,0	86,0±2,6	80,3±2,9	65,4±3,0	47,9±5,0
One-Class SVM и GRU+CNN	93,5±1,5	86,9±2,2	81,2±3,4	69,8±3,3	51,9±3,8

Из результатов видно, что сохраняется тенденция к низким результатам статистических подходов. Использование метода One-Class SVM, способного детектировать аномалии, в сочетании с комбинацией GRU+CNN показало лучшие результаты во всех экспериментах. Подход на основе порогового значения SoftMax уступает ему в точности от 1 до 4%.

3.7 Оценка разработанных методик в сравнении с аналогами

Для оценки практической значимости разработанных методик было проведено сравнение точности с аналогичными исследованиями, направленными на определение автора русскоязычного текста.

Результаты проведенного сравнения для художественных текстов приведены в таблице А.1, для коротких сообщений пользователей социальных сетей – в таблице А.3 (приложение А).

Предложенные методики:

а) способны идентифицировать автора текста с высокой точностью при наличии 3 обучающих образцов по 15000 символов в случае длинных текстов и 40 образцов до 250 символов для коротких текстов;

б) могут решать все типы задач идентификации автора текста, включая случаи открытого и закрытого множества авторов, верификацию авторства с высокой точностью, достигающей 90–99%;

в) применимы для художественных, любительских и текстов социальных сетей и мессенджеров;

г) апробированы на корпусах, включающих от 2 до 50 авторов художественных и любительских текстов, сообщениях социальных сетей и мессенджеров.

В разработанных методиках используется широкий спектр современных технологий и алгоритмов, включая SVM, отбор признаков регуляризационным алгоритмом, гибридную нейросетевую модель GRU+CNN, сочетающую сверточные и рекуррентные слои. Это позволяет эффективно обрабатывать и анализировать тексты, учитывая как стилистические, так и семантические особенности различных типов текстов.

В трудах российских исследователей не рассматривались задачи определения авторства любительских текстов и случай определения автора для открытого множества кандидатов. Приведенные рекомендации по применению аналогичных подходов включают использование большего объема обучающих данных. При этом их результаты уступают предложенной методике от 1% до 44%.

3.8 Выводы по главе

1. Сформулированы и решены задачи идентификации автора в случае открытого и закрытого множества кандидатов, верификации авторства, рассмотрены монотематический и кросс-тематический случай, а также влияние объема и количества выборок, количества потенциальных авторов в случае профессиональной

деятельности, любительского написания текстов и общения в социальных сетях и мессенджерах.

2. Рассмотрена возможность применения классических статистических и современных нейросетевых методов, в том числе глубоких архитектур и их сочетаний, а также методов отбора информативных признаков для идентификации автора текста.

3. Для идентификации автора в случае закрытого множества кандидатов лучшими были признаны GRU+CNN и SVM с регуляризационным отбором информативных признаков. Результаты достигают 98,6%, 98,9%, 96,3% для художественных, любительских и коротких текстов, соответственно. При тестировании методики для случая 50 классов результаты составили 56,9%, 62,4%, 56% для тех же типов текстов.

4. Для идентификации автора в случае открытого множества кандидатов лучшей была признана комбинация One-Class SVM и GRU+CNN. Точность методики достигает 94,2%, 98,2% и 93,5% для художественных, любительских и коротких текстов, соответственно. При увеличении количества классов до 50 для тех же типов текстов получены результаты 40,8%, 45,3%, 51,9%.

5. Проведена серия экспериментов, направленных на использование One-Class SVM для верификации автора текста. Точность верификации авторов художественных, любительских и коротких текста достигает 99,2%, 98,9% и 97,8%, соответственно.

6. Установлено, что применение генетических алгоритмов и регуляризационного отбора позволяет улучшить точность идентификации автора на 16%. Однако их применение увеличивает временные затраты более чем в 10 раз. При схожей постановке задачи предпочтительным являет использование глубоких нейронных сетей.

7. Рекомендуемый объем текста для обучения моделей в случае художественных и любительских текстов составляет 3 образца по 15000 символов. В случае коротких текстов для обучения следует использовать 40 текстов каждого автора.

8. Установлено, что в случае атаки, проводимой с помощью внедрения образцов, созданных современными генеративными нейросетями, методика достигает точности 90%.

9. Итоговая методика включает GRU+CNN с параметрами: размерность входного вектора – 200, количество сверточных фильтров – 4096, размер ядра Conv1D – 3, функция активации Conv1D – ReLU, использование GlobalMaxPooling, количество нейронов полносвязного слоя – 256 (функция активации: ReLU), количество нейронов слоя GRU – 128; SVM с параметрами: алгоритм обучения – sequential optimization method, ядро – линейное, параметр регуляризации – 0,1, допустимый уровень ошибки – 0,00001, нормирование и эвристика сжатия – включено; One-Class SVM с параметрами: линейное ядро, коэффициент регуляризации – 0,1, допуск по критерию останова – 0,00001.

4 МЕТОДИКА ИДЕНТИФИКАЦИИ АВТОРА ИСХОДНОГО КОДА ПРОГРАММЫ

Идентификация автора исходного кода [125, 259] – это проблема определения истинного автора исходного кода на основе образцов текстов программ, написанных авторами-программистами. Эффективный процесс идентификации подразумевает выявление признаков, характеризующих стиль, приемы и привычки написания кода, в том числе нарушающие парадигмы языков программирования.

Формализовать задачу определения автора исходного кода программы можно следующим образом. Пусть имеется множество авторов $\mathbf{A} = \{a_1, a_2, ..., a_l\}$ и множество исходных кодов программ $\mathbf{S} = \{s_1, s_2, ..., s_k\}$, где каждый исходный код рассматривается как вектор признаков $\mathbf{X} = \{x_1, x_2, ..., x_n\}$. Для подмножества исходных кодов $\mathbf{S}' = \{s_1, s_2, ..., s_m\} \subseteq \mathbf{S}$, где $m<k$ – авторы известны, существует множество пар «автор-исходный код» $(s_i, a_j) \in \mathbf{D} \subseteq \mathbf{S}' \times \mathbf{A}$, где $s_i \in \mathbf{S}'$, $a_j \in \mathbf{A}$. Решение задачи состоит в идентификации автора, принадлежащего множеству **A**, являющегося истинным автором анонимных образцов исходных кодов подмножества $\mathbf{S}'' = \mathbf{S} / \mathbf{S}'$.

Данную задачу следует рассматривать как задачу многоклассовой классификации. Так, множество **A** состоит из множества предопределенных классов и их меток, множество **D** включает в себя исходные коды для обучения классификатора, а множество $\mathbf{S}''$ содержит классифицируемые исходные коды.

Цель состоит в разработке модели, решающей поставленную задачу – нахождении целевой функции $F : \mathbf{S} \times \mathbf{A} \to [0;1]$, которая относит некоторый исходный код из множества **S** к его истинному автору. Значение функции описывается как степень принадлежности объекта классу, где 1 соответствует полное соответствие целевому классу, а 0, напротив, полное несоответствие.

4.1 Выбор подходов к идентификации автора исходного кода программы

Современные решения, направленные на определение автора исходного кода программы, зачастую не учитывают особенности создания современных программных продуктов. К таким особенностям относятся:

1. Обфускация – процесс, направленный на запутывание исходного кода программы без нарушения его функционирования с целью сокрытия личности программиста и снижения удобочитаемости кода.

2. Стандарты кодирования – создание кода в соответствии с правилами и соглашениями, принятыми сообществом или командой разработчиков.

3. Командная разработка – процесс написания исходного кода программы двумя и более разработчиками.

4. Владение двумя и более языками – использование программистом в рамках одного проекта двух и более языков программирования.

5. Искусственная генерация кода – использование генеративных моделей НС для генерации фрагментов исходного кода или всего исходного кода.

Обозначенные особенности и специфика исходных кодов как объекта исследования должны учитываться при выборе подхода к идентификации автора. Наиболее подходящими являются подходы, во-первых, работающие на основе информативных признаков, отобранных экспертным путем, во-вторых, осуществляющие самостоятельные извлечение и анализ признаков, характеризующих автора.

В качестве алгоритмической основы методики рассматриваются современные модели, демонстрирующие высокую эффективность при решении смежных задач: fastText, BERT и SVM с экспериментально подобранными признаками. Выбор этих моделей обусловлен тем, что они являются state-of-the-art (SOTA) моделями для решения задач анализа искусственно-языкового текста, что подтверждается результатами исследователей в подразд. 1.4.

Для обеспечения высокой эффективности модели fastText требуется настройка ее параметров при помощи встроенных средств автотюнинга.

В результате автоматического подбора были определены оптимальные параметры fastText: скорость обучения – 0,05, частота обновления скорости – 50, максимальный размер *n*-грамм – 5, размер контекстного окна – 5, количество отрицательных образцов – 3, число эпох – 100. Остальные параметры были установлены по умолчанию.

В качестве предобученных BERT были выбраны две модели: Bert-base-multilingual-case, являющаяся базовой для решения мультиязычных задач анализа текста, и CodeBERT [52], разработанный для решения задач, связанных с анализом исходных кодов программ на различных языках. Оптимальные параметры обеих моделей были подобраны при помощи жадного поиска. По результатам поиска для BERT были выбраны настройки: скорость обучения – 0,00003, коэффициент регуляризации – 0,01, размер батча – 16, число эпох обучения – 10. По результатам поиска для CodeBERT выбраны: скорость обучения – 0,00003, коэффициент регуляризации – 0,015, размер батча – 16, число эпох обучения – 10. Стоит отметить, что стандартная архитектура BERT (и CodeBERT, соответственно), приведенная в п. 2.7.4, модифицируется под задачу классификации путем добавления 1 слоя прореживания с коэффициентом 0,1 для противодействия переобучению и выходного слоя Softmax для получения распределения вероятностей по классам (рисунок 4.1).

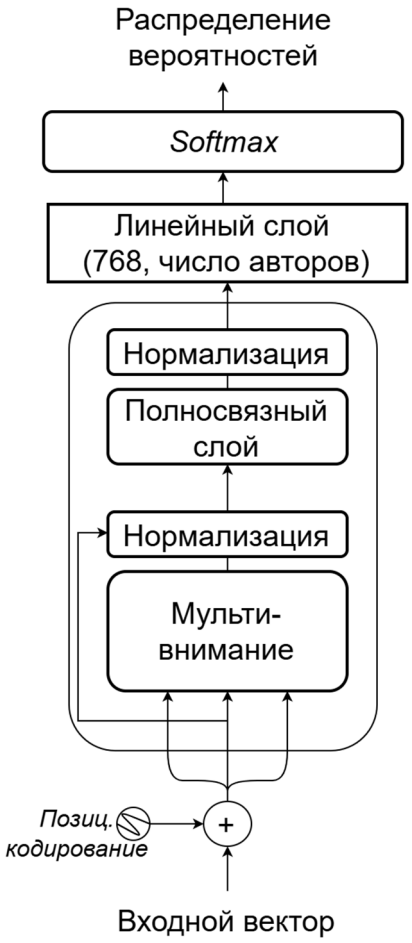


Рисунок 4.1 – Архитектура классификатора на основе CodeBERT

Для обучения SVM, как модели машинного обучения, требуется предопределенное множество информативных признаков:

1. Лексические признаки (таблица 4.1). Такие признаки демонстрируют приверженность автора к использованию тех или иных конструкций, статистику использования ключевых слов, операторов, циклов, а также определенного стиля именования переменных и функций.

Таблица 4.1 – Множество лексических признаков

Признак	Описание
nestingDepth	Наивысшая степень вложенности операторов и циклов
MacrosToLength	Отношение числа директив препроцессора к длине кода
avgLineLength	Средняя длина строки
CommentsToLength	Отношение числа комментариев к длине кода
LiteralsToLength	Отношение числа литералов к длине кода
TernaryToLength	Отношение числа вхождений тернарного оператора к длине кода
FunctionsToLength	Отношение числа функций к длине кода
TokensToLength	Отношение числа токенов к длине кода
avgParams	Среднее число параметров среди всех функций
keywordToLength	Отношение числа ключевых слов к длине кода

2. Структурные признаки (таблица 4.2). Такие признаки демонстрируют стиль автора в части использования пробельных символов и оформления блоков кода. Последнее особенно важно в тех языках программирования, пробельные символы в которых не являются инструкцией для компилятора или интерпретатора и могут применяться произвольно на усмотрение программиста. Примерами таких языков являются *C*-подобные языки программирования.

Таблица 4.2 – Набор структурных признаков

Признак	Описание
newLineBeforeOpenBrace	Бинарное значение, определяющее, предшествуют ли большинству фигурных скобок символы новой строки
tabsLeadLines	Бинарное значение, определяющее, начинается ли большинство строк кода с отступов
EmptyLinesToLength	Отношение числа пустых строк к длине кода
whiteSpaceRatio	Соотношение между числом пробельных символов и числом не пробельных символов
TabsToLength	Отношение числа символов табуляции к длине кода
SpacesToLength	Отношение числа символов пробела к длине кода

3. Признаки, определяющие нарушение парадигм программирования (таблица 4.3). Такие признаки (так называемые «запахи кода») отражают несоответствие кода программиста понятиям и идеям, определяющим стиль написания кода на определенном языке.

Таблица 4.3 – Набор «запахов кода»

Признак	Описание
dataClass	Отношение числа классов, содержащих исключительно атрибуты и методы get и set, к общему числу классов
featureEnvy	Отношение числа методов, обращающихся к данным другого класса чаще, чем к собственным, к общему числу методов
longMethod	Среднее количество строк в методах
classMediator	Отношение числа классов, делегирующих большую часть методов другому классу, к общему числу классов
switchLeadStat	Отношение числа операторов Switch к общему числу условных операторов
longMethodShare	Отношение числа длинных методов к общему числу методов
longParamList	Среднее число параметров в методах классов

Описанное множество признаков позволяет анализировать исходный код или фрагмент кода даже в том случае, когда он содержит синтаксические или семантические ошибки, то есть не требует компилируемости.

Важно учитывать, что множество является избыточным и должно быть сокращено путем фильтрации малоинформативных для конкретного случая признаков. Для осуществления эффективной фильтрации используется FCBF (подразд. 2.9).

4.2 Сложные случаи идентификации автора исходного кода

4.2.1 Обфускация исходного кода

Обфускация исходного кода [16, 41, 175] – это процесс приведения кода программы в вид, осложняющий анализ, но сохраняющий функциональность. Как правило, обфускация выполняется с целью защиты коммерческого ПО от плагиата, а также защиты инкогнито автора-программиста. В самом простом случае обфускация состоит в изменении наименований идентификаторов, изменении стиля оформления блоков кода. Однако современные инструменты позволяют выполнять более сложные преобразования: внедрение псевдокода, дубликатов, функций-пустышек и т.п.

Проблема идентификации автора обфусцированного кода редко рассматривается исследователями в работах, посвященных анализу программного кода на авторство. Исходя из этого, было решено произвести оценку влияния обфускации на эффективность подходов.

Для интерпретируемого языка JavaScript использовались JS Obfuscator Tool [115] и JS-obfuscator [116], для Python – PyArmor [178] и Opy [162], для PHP – Yakpro-po [219] и PHP Obfuscator [170]. Для компилируемого языка C++ использовался Cpp Guard [56], для C – Analyse C [15].

Обфускация образцов исходных кодов на языке программирования JavaScript (JS) выполнялась при помощи JS Obfuscator Tool и JS-obfuscator. В процессе работы инструмента JS Obfuscator Tool происходят простые лексические модификации: изменение наименований классов, методов и переменных, удаление пробельных символов и комментариев и преобразование строк в шестнадцатеричные последовательности с целью последующего их кодирования 64 символами ASCII. Инструмент «JS-obfuscator» имеет другой принцип работы и выполняет, помимо перечисленных преобразований, запутывание PHP, HTML и др. вставок.

Для обфускации кода на языке Python использовалась PyArmor и Opy. Первый инструмент запутывает код функций и методов в процессе интерпретации

путем модификации байт-кода каждого объекта, а также удаления локальных переменных после выполнения методов и функций. Второй инструмент осуществляет простые лексические преобразования, включающие замену имен методов, функций, классов, переменных и констант на последовательности символов «1» и «l».

Образцы на языке PHP обфусцировались с помощью Yakpropo и PHP Obfuscator. Данные инструменты выполняют простые преобразования, направленные на удаление комментариев и пробельных символов, а также случайное переименование констант, переменных, методов, функций, классов.

Для компилируемых языков *C* и *C++* использовались обфускаторы CPP Guard и AnalyzeC. Первый инструмент нацелен на удаление комментариев, пробельных символов, добавление псевдокода, не влияющего на вычислительную сложность программы, обработку препроцессорных директив, замену наименований переменных и функций на последовательность случайных символов, а также преобразование строк кода в их шестнадцатеричное представление.

4.2.2 Использование стандартов кодирования

Стандарты кодирования являются еще одной проблемой при идентификации автора программного кода [214]. Современное сообщество разработчиков, а также многие компании, занимающиеся разработкой заказного ПО, прибегают к использованию стандартов кодирования. Такие стандарты определяются правилами, соглашениями и рекомендациями относительно оформления исходного кода и необходимы для оптимизации времязатрат на чтение и анализ кодов, написанных другими разработчиками.

Стандартам кодирования посвящено большое количество исследований [133, 92]. Однако их авторы уделяют недостаточно внимания тому, как стандарты влияют на авторский стиль разработчика. В частности, на его профессиональные приемы и привычки, несущие в себе множество информативных признаков, позво-

ляющих различать авторов между собой. Оценка этого влияния является необходимой для решения задачи идентификации автора исходного кода.

Для проведения экспериментов по идентификации автора исходного кода, написанного в соответствии со стандартами кодирования, были собраны данные из репозиториев разработчиков проекта Linux Kernel [134]. Общее количество исходных кодов в наборе данных составило свыше 250 тыс. экземпляров, написанных на языках программирования *C*, *C++* и языке ассемблера пятью наиболее активными разработчиками.

4.2.3 Использование нескольких языков программирования

Важной проблемой современных исследований, посвященных задаче идентификации автора исходного кода программы, является отсутствие экспериментов, направленных на оценку точности разработанных решений применительно к смешанным наборам данных. Зачастую реализация продукта происходит на нескольких языках программирования одновременно. Набор языков программирования, используемых в рамках разработки продукта, может различаться и называется его стеком. Точно такой же стек существует и у каждого разработчика, то есть группа языков, на которых он осуществляет разработку исходных кодов программ или их модулей.

Реализация основной части программы нередко осуществляется на одном языке, а необходимость в другом возникает лишь для создания вспомогательных инструментов. Однако возникают ситуации, когда разработка выполняется в равной степени на двух или даже трех языках программирования. Исходя из этого, количество данных, собираемых с целью обучения модели идентификации автора исходного кода программы, может оказаться недостаточным для получения разработанным инструментом достоверного результата. Отсюда возникает необходимость в оценке способности предложенной методики демонстрировать сопоставимый однородным данным результат применительно к неоднородным. В данном

случае, реализованным на различных языках программирования исходным кодам программ.

Проверке подлежит гипотеза о слабой зависимости авторского стиля от языка программирования, на котором пишет программист, и свободном переносе информативных признаков с одного языка на другой. Модель, лежащая в основе методики идентификации автора исходного кода, должна обладать способностью к поиску и анализу как явных, так и неявных авторских признаков даже на основе данных, включающих в себя образцы на различных языках программирования.

Для получения обучающих наборов, включающих в себя смешанные данные, был осуществлен поиск авторов-программистов, являющихся контрибьюторами проектов на разных языках программирования. Извлекались исходные коды программ, написанные конкретным автором и размещенные в соответствующем языку репозитории. При формировании набора данных разделение образцов исходных кодов программ не осуществлялось.

4.2.4 Искусственно-сгенерированные исходные коды

Современные модели, предназначенные для создания текстов с нуля или на основе контекста, демонстрируют высокую эффективность [69, 81, 131, 172].

Генерация естественно-языковых текстов, зарекомендовавшая себя в различных сферах деятельности человека, спровоцировала появление нового направления – генерацию искусственно-языковых текстов, в том числе исходных кодов программ. Модели, позволяющие создавать исходный код самостоятельно или дописывать текст программы за пользователя, позволяют сократить временные затраты программистов путем избавления их от написания рутинного кода, не требующего исследовательского или творческого вклада.

Методика идентификации автора искусственно-сгенерированного исходного кода программы должна не только эффективно выявлять информативные признаки авторов-программистов, но также быть способной, как минимум, разделять

авторство между различными генеративными моделями, а следовательно, выявлять отличительные черты создания кода искусственным путем.

Разработчики решений, предназначенных для генерации исходных кодов программ, также учли положительный опыт применения GPT-моделей. На их основе появились наиболее известные инструменты Open AI Codex, GitHub Copilot, AlphaCode, JARVIS Sber AI, PolyCoder, CodeBERT [11, 52, 88, 161, 174, 192]. Несмотря на то, что большинство существующих решений в области генерации исходного кода, демонстрируют впечатляющие результаты, их применение в рамках данного исследования невозможно, так как они являются закрытыми для сообщества разработчиков. Однако все рассмотренные инструменты базируются на моделях семейства GPT. Общедоступным полноценным решением является только CodeBERT. AlphaCode не является подходящим в связи с тем, что предназначено исключительно для создания кода для соревнований. Таким образом, в рамках данного исследования решено использовать GPT-2, GPT-3, GPT-4 и CodeBERT.

4.2.5 Коммиты исходных кодов программ

Наиболее распространенным среди сложных случаев анализа неоднородных данных является идентификация автора исходного кода программы на основании его коммитов – последних изменений исходного кода в репозиторий. Так как инструменты определения авторства программ считаются наиболее востребованными для коммерческой среды, следует учитывать специфику работы этой области. Современные IT-компании пользуются различными технологиями для гибкой совместной разработки программных продуктов и используют в своей ежедневной деятельности различные системы контроля версий.

Согласно принципам групповой разработки, большинство программ и программных комплексов создаются не единолично одним программистом, а целой командой. Для решения задачи идентификации автора исходного кода важно уметь

правильно извлекать образцы конкретных авторов-программистов, а значит, разграничивать коммиты в соответствии с именами загрузивших их пользователей.

В данном случае, для создания обучающего набора данных используется информация о коммитах отдельных пользователей в репозитории.

4.3 Предварительная оценка подходов к идентификации автора исходного кода программы

Обучение моделей и их дальнейшая оценка подразумевают использование экспериментальных данных. Полученный с хостинга IT-проектов набор данных (см. п. 6.7.2) позволяет осуществлять эффективное обучение и объективную оценку моделей за счет разнообразия входящих в него образцов исходных кодов.

В качестве процедуры оценки моделей использовалась перекрестная проверка по 10 частям, решающая распространенную проблему чрезмерного разброса оценок. В качестве метрики качества используется точность. Результаты оценки моделей на наборе данных 10 авторов для всех архитектур (рисунок 4.2).

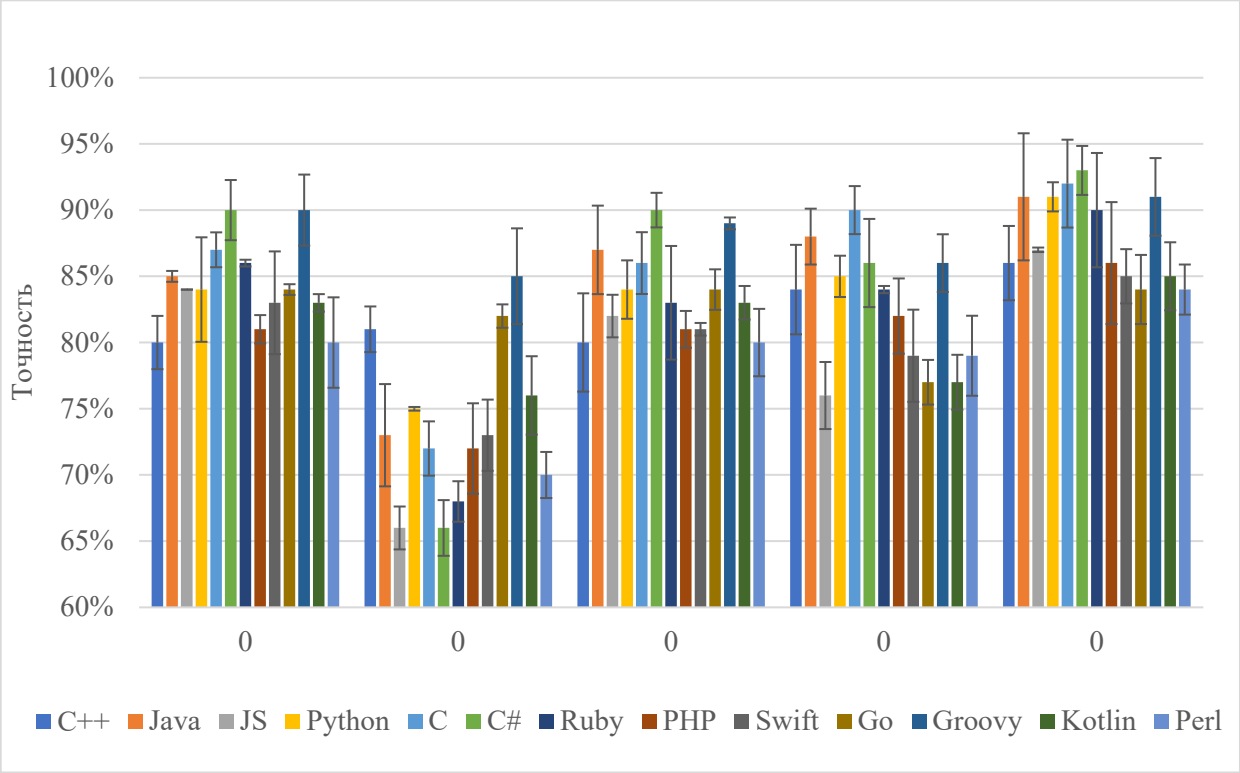


Рисунок 4.2 – Результаты экспериментов для разных моделей

Согласно полученным результатам, лучшая точность для всех языков программирования была достигнута моделью CodeBERT. Мультиязыковой BERT, fastText и SVM+FCBF уступают в точности до 5%. Это обусловлено двумя факторами. Во-первых, CodeBERT имеет специализированные знания, полученные в результате ее обучения на большом наборе естественно- и искусственно-языковых данных на различных языках. Во-вторых, CodeBERT использует специальные методы токенизации данных, направленные на эффективную работу с особенностями синтаксиса и структуры кода.

На основании полученных метрик, в качестве алгоритмической основы разрабатываемой методики было решено использовать CodeBERT, точность классификации авторов которым достигает 93%.

4.4 Методика идентификации автора исходного кода программы на основе CodeBERT

Методика идентификации автора исходного кода [22, 53, 124, 249, 260, 261] включила в себя все шаги, необходимые для принятия моделью решения относительно принадлежности анонимного образца исходного кода одному из тренировочных классов (рисунок 4.3). Алгоритмом принятия решений является глубокая НС, способная самостоятельно выделять информативные признаки из не предобработанного текста в векторном виде и не нуждающаяся в признаковом пространстве.

Методику составляют следующие шаги:

1. Токенизация данных для обучения и оценки модели. На данном шаге выполняется преобразование исходного кода в последовательности токенов и их векторизация с использованием токенизатора CodeBERT на базе Roberta. Также производится преобразование меток авторов в числовые представления.

2. Обучение и оценка модели. Модель обучается на протяжении заданного числа эпох. На каждом шаге CodeBERT делает предсказания для текущего батча данных, вычисляется функция потерь, затем градиенты этой функции исполь-

зуются для обновления параметров модели. После каждой завершенной эпохи модель проверяется на тестовом наборе.

3. Токенизация анонимного образца исходного кода. Анонимный образец кода разбивается на токены и преобразуется в векторный вид.

4. Определение авторства анонимного образца обученной моделью. Обученный CodeBERT делает предсказание для анонимного образца. За счет применения Softmax, на выходе модель выдает распределение вероятностей по всем авторам-кандидатам. Наибольшая вероятность указывает на истинного автора кода.

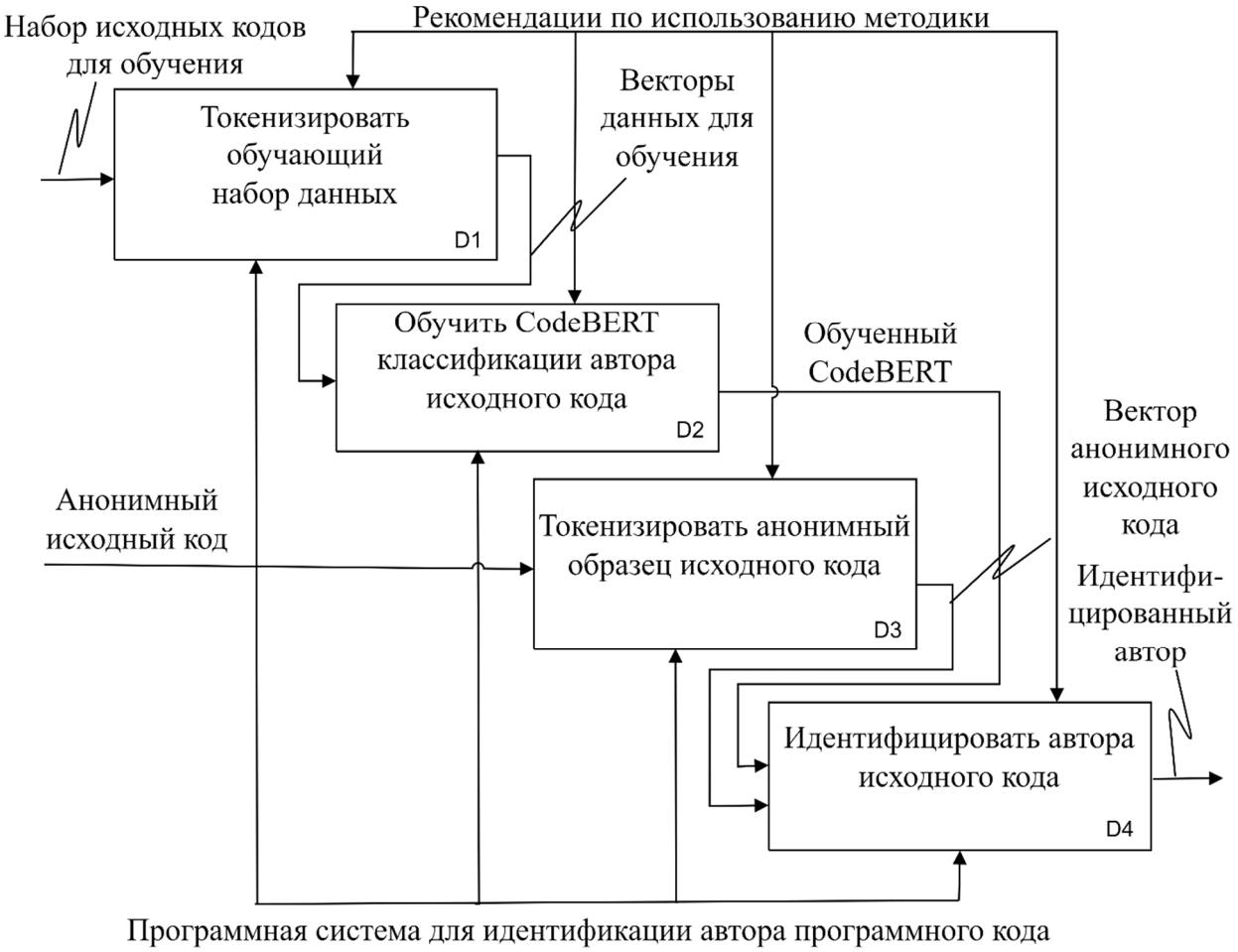


Рисунок 4.3 – Обобщенная методика идентификации автора исходного кода

4.5 Результаты экспериментов

Процесс проведения экспериментов представлен на рисунке 4.4. Для обучения и оценки модели CodeBERT был создан набор данных, охватывающий все сложные случаи. Образцы кодов из набора соответствуют следующим критериям:

– длина кода не менее 1000 символов;

– отбирается не более 30 образцов исходного кода одного автора;

– выбор образца случайный.

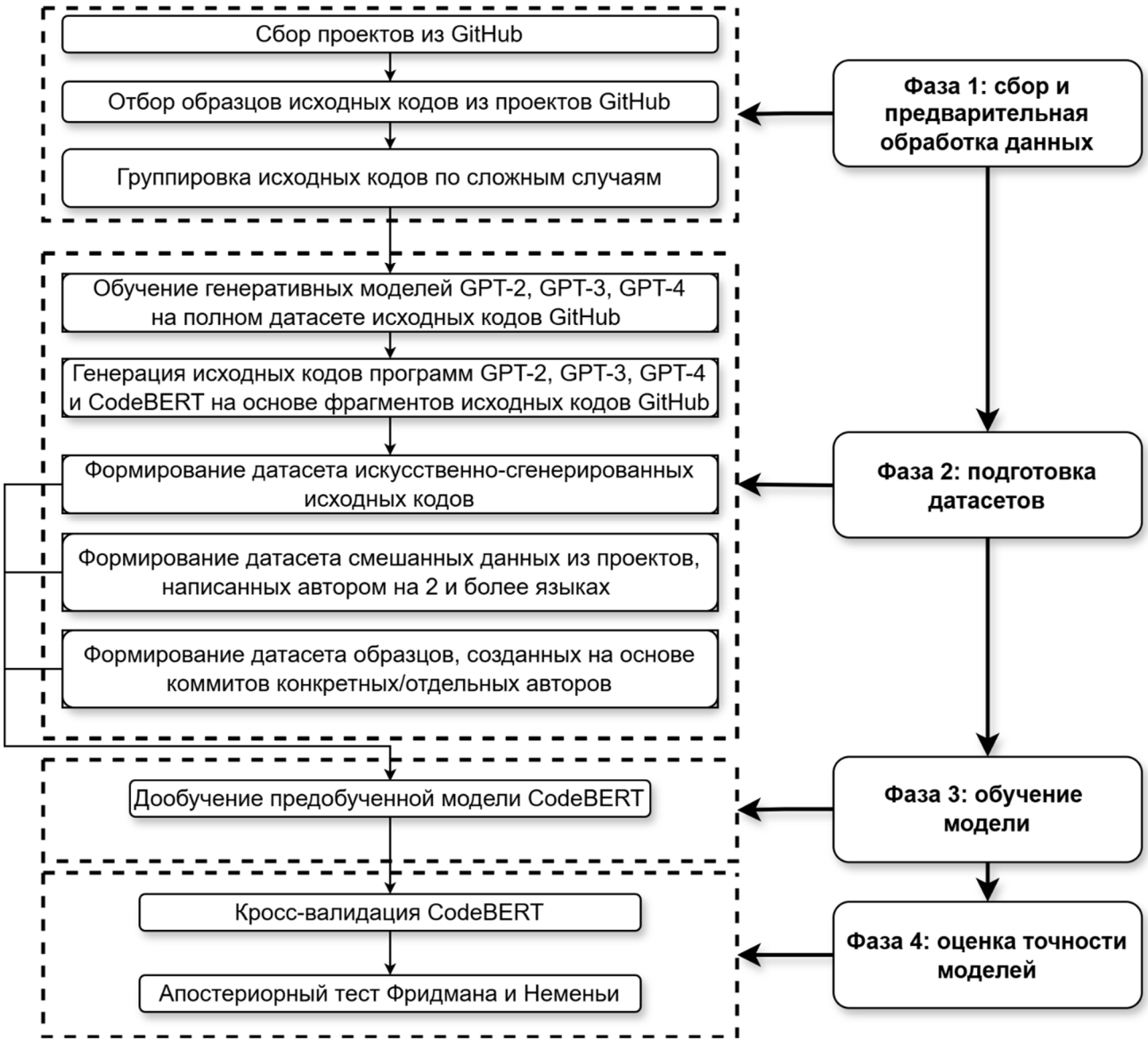


Рисунок 4.4 – Процесс проведения экспериментов для сложных случаев

Согласно критериям, минимальное количество строк кода для обучения одного автора – 90, максимальное – 90000. Эти пределы были выбраны на основе статистики собранного набора данных. Максимальное количество эпох для обучения модели – 10. В этом исследовании была применяется процедура «Early Stopping», автоматически прерывающая обучение. Для оценки точности модели использовался метод перекрестной проверки по 10 блокам.

4.5.1 Результаты экспериментов для простых случаев идентификации автора исходного кода

Первым шагом стала оценка методики на различных по количеству авторов корпусах исходных кодов. Для эксперимента были использованы корпуса размером 5, 10 и 20 авторов. Результаты представлены на рисунке 4.5.

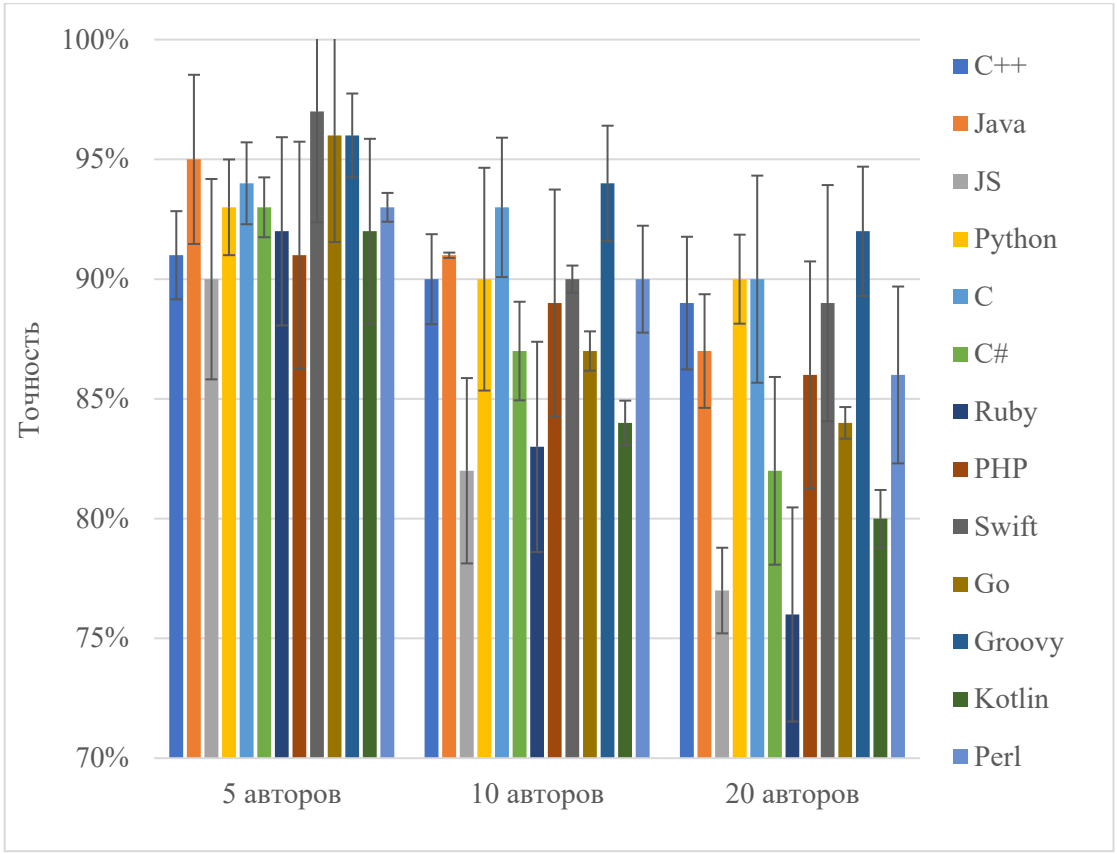


Рисунок 4.5 – Результаты экспериментов для простых случаев идентификации автора исходного кода

Обращаясь к анализу подходов к идентификации автора исходного кода, можно сделать вывод о том, что предложенная методика демонстрирует конкурентные результаты для всех языков программирования. Более того, она оказалась высокоэффективной и для языков программирования, не рассмотренных в научных трудах исследователей – C, C#, Ruby, PHP, Swift, Kotlin, Go, Groovy, Perl. Особенности языка, на котором пишет программист, влияют на результативность, но в целом методику можно считать универсальной.

4.5.2 Результаты экспериментов для сложных случаев идентификации автора исходного кода

В качестве сложных рассмотрены случаи обфускации кода, написания кода согласно стандартам кодирования, обучения модели на смешанном наборе данных, искусственной генерации кода и совместной разработки.

На основе экспериментальных данных были сформированы обучающие наборы, состоящие из 10, 20 и 30 случайных файлов длиной не менее 1000 и не более 10000 символов. Результаты экспериментов для исходных кодов, написанных в соответствии со стандартами кодирования, представлены на рисунке 4.6.

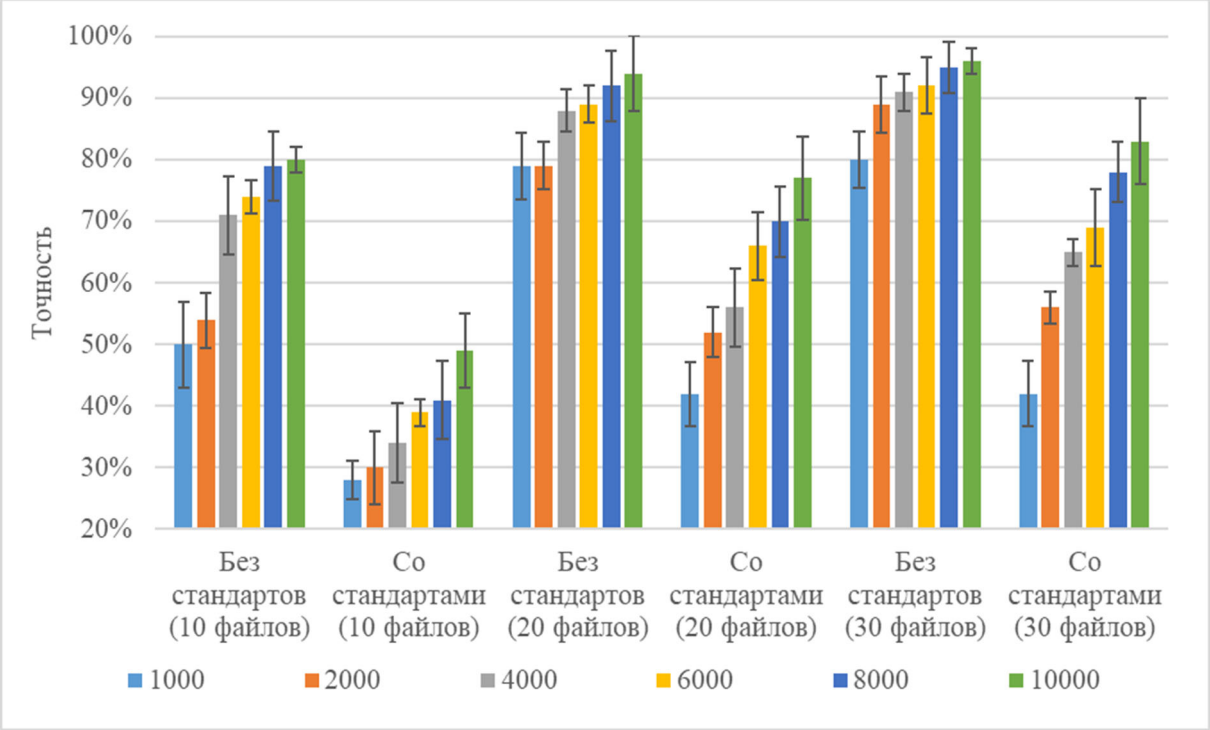


Рисунок 4.6 – Результаты эксперимента со стандартами кодирования

Исходя из полученных результатов, можно сделать вывод о негативном влиянии стандартов кодирования на процесс идентификации автора. Такой результат получается в результате сглаживания таких проявлений авторского стиля как структура логических блоков, именование идентификаторов, оформление комментариев, предпочтение определенного типа пробельных символов и т.п. В результате применения стандартов кодирования исходный код изменяется до такой степени,

что работа методики на основе CodeBERT сводится к поиску неподконтрольных программисту признаков, например, связанных с его опытом и квалификацией. Тем не менее, при наличии достаточного объема обучающих данных методика позволяет добиться приемлемой для решения прикладных задач точности в 80%.

Для случая обфускации на рисунке 4.7 представлены результаты экспериментов с интерпретируемыми и компилируемыми языками программирования.

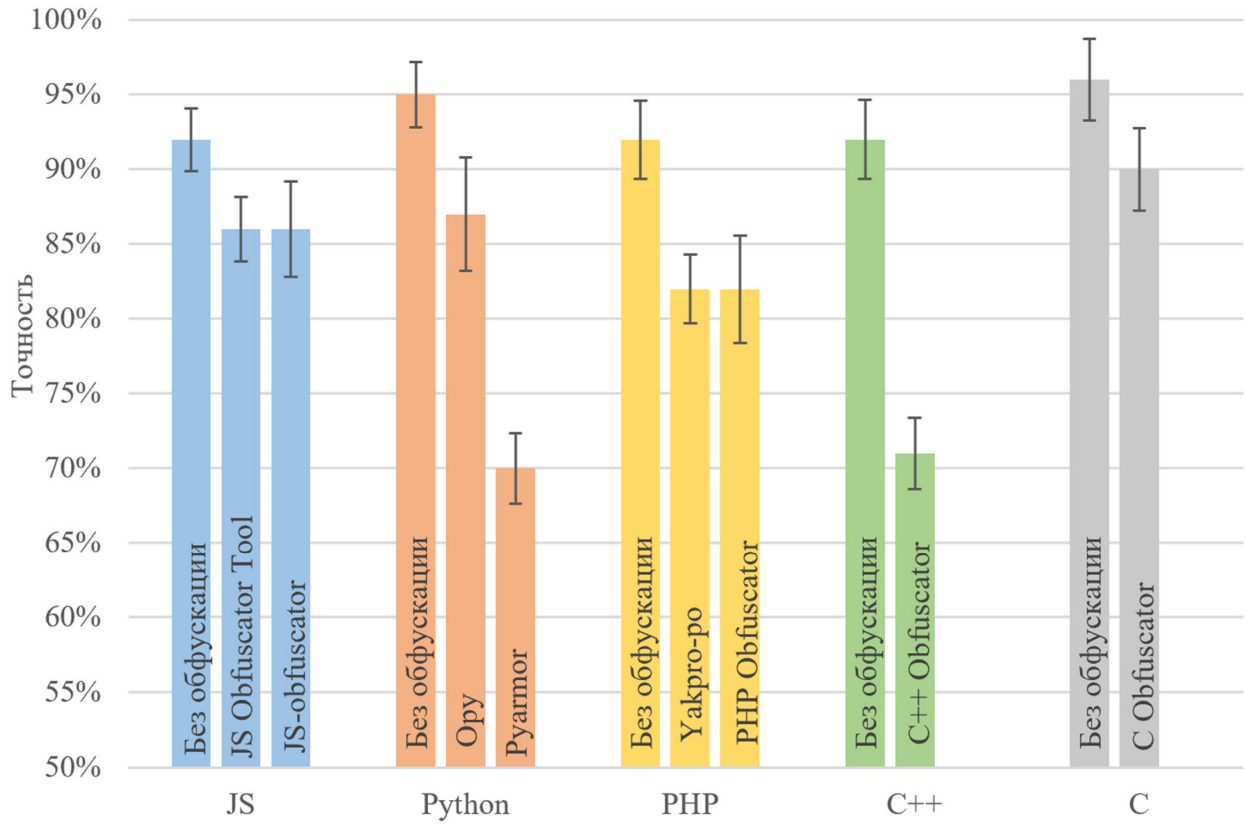


Рисунок 4.7 – Результаты экспериментов для случая обфускации

Простая лексическая обфускация не оказывает существенного влияния на эффективность методики, в то время как более сложные преобразования снижают точность в среднем на 20–30%.

Исходя из полученных результатов, CodeBERT является устойчивым к простым лексическим преобразованиям инструментами обфускации. Средняя для различных языков и обфускаторов потеря в точности составляет 8%. Более сложные модификации исходного кода, подразумевающие изменение байт-кода объектов, внедрение псевдокода и т.п. приводят к существенным потерям в точности, достигающим 30% при недостаточном числе обучающих образцов.

Согласно результатам проведенных экспериментов, средняя точность классификации для авторов, использующих два языка программирования, достигает 92% (рисунок 4.8). В то же время следует отметить, что точность определения автора по языковым парам, имеющим сходство в синтаксисе, превышает точность языковых пар, не имеющих сходства. Этот вывод верен для языковых пар JavaScript-C++, Swift-Java и JavaScript-Python. Средняя точность для авторов, разрабатывающих на трех и более языках программирования, достигает 91% (рисунок 4.9).

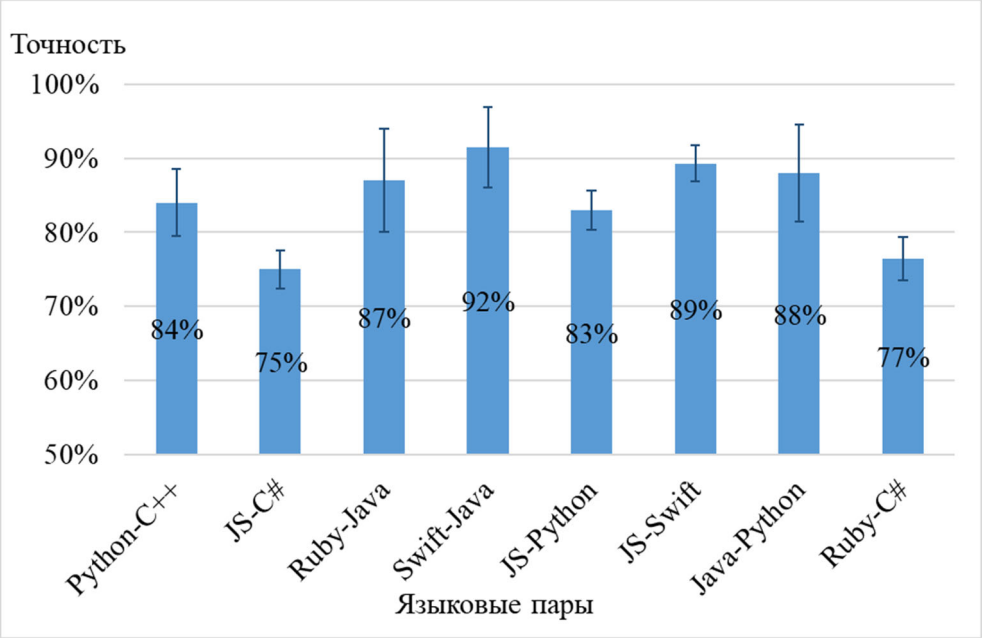


Рисунок 4.8 – Результаты экспериментов с языковыми парами

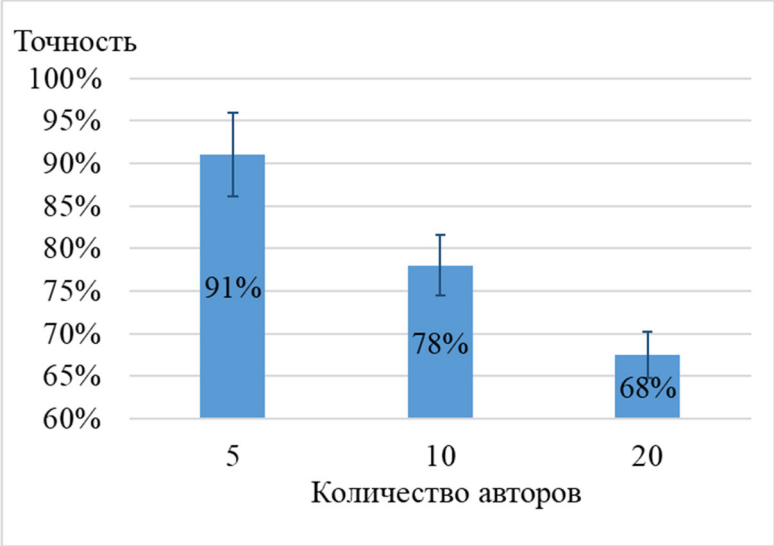


Рисунок 4.9 – Результаты экспериментов со смешанным набором данных

Поскольку наборы данных полностью смешаны и могут включать более десяти языков программирования на одного автора, невозможно оценить уровень важности каждого языка. Следует отметить, что этот случай представляет особый исследовательский интерес, поскольку на практике встречается крайне редко.

В рамках оценки точности идентификации автора искусственно-сгенерированного исходного кода программы было рассмотрено два сложных случая. Первый и самый простой из них – разграничение авторства между человеком и генеративной моделью (рисунок 4.10). В данном случае оценка производилась следующим образом. В каждом эксперименте для 5, 10 и 20 авторов одним из авторов являлась генеративная модель. В исследованиях участвовали образцы, написанные авторами-программистами, и образцы, сгенерированные моделями семейства GPT: GPT-2, GPT-3, GPT-4, CodeBERT. Все образцы были реализованы на наиболее популярном языке Java. Средняя точность идентификации кода, созданного GPT-2 составила 96%, GPT-3 и GPT-4 – 92% и CodeBERT – 90%. Разница в точности обусловлена тем, что модели 3 и 4-го поколений демонстрируют лучшую генеративную способность, чем 2-го, а CodeBERT является специализированной моделью для исходных кодов. Следовательно, разделительная способность модели идентификации автора также снижается.

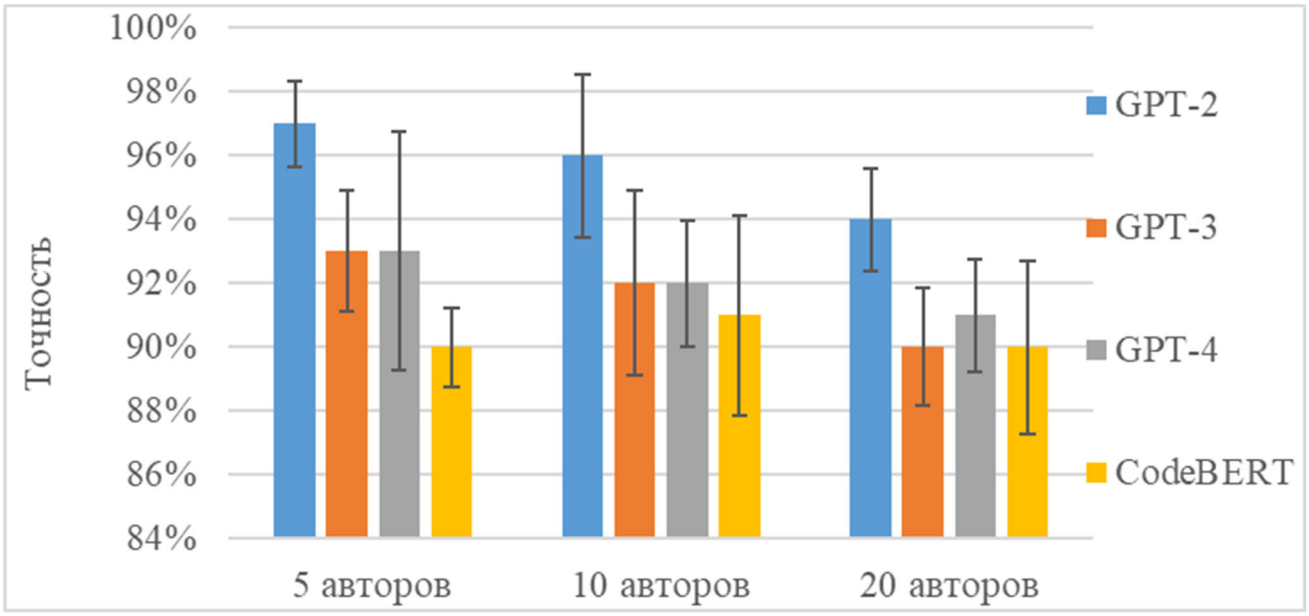


Рисунок 4.10 – Результаты эксперимента по разделению авторства между человеком и генеративной моделью

Второй эксперимент, посвященный искусственно-сгенерированным исходным кодам, был нацелен на оценку точности модели при разграничении авторства между четырьмя различными генеративными моделями (рисунок 4.11).

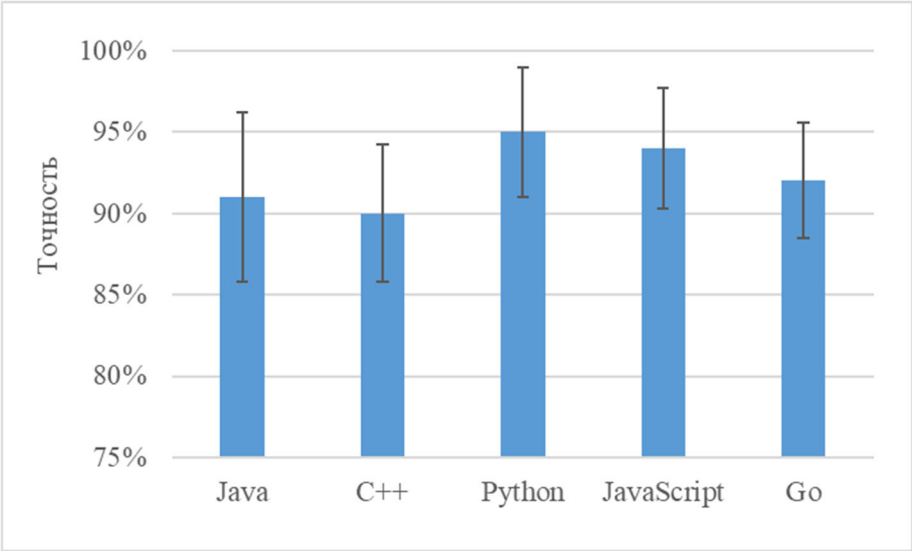


Рисунок 4.11 – Результаты эксперимента по определению автора-модели

В данном случае эксперимент был поставлен таким образом, что в обучении участвовали 4 автора-модели: GPT-2, GPT-3, GPT-4 и CodeBERT. Целью работы методики являлось определение принадлежности анонимного исходного кода, также сгенерированного искусственно, к одной из языковых моделей. Эксперимент проводился для 5 языков программирования: Java, C++, Python, JavaScript и Go. Точность определения авторства исходного кода, сгенерированного языковой моделью, для всех языков программирования составила более 90%.

Последнее исследование было посвящено идентификации автора изменений, сделанных разработчиками в одном и том же проекте (рисунок 4.12). Набор данных формировался на основе образцов, содержащих коммиты (фрагменты, измененные в общем файле исходного кода конкретным пользователем) отдельных пользователей.

С целью разработки методических рекомендаций по применению методики для идентификации авторов-программистов, разрабатывающих коммерческое ПО и использующих системы контроля версий, было решено провести эксперимент для разного количества файлов – 5, 10 и 20, соответственно. Согласно результатам,

для получения высокоточного (90%) инструмента принятия решений об авторстве следует использовать не менее 10 файлов исходного кода каждого автора. При несоблюдении данного условия точность может быть недостаточной для эффективного решения задачи.

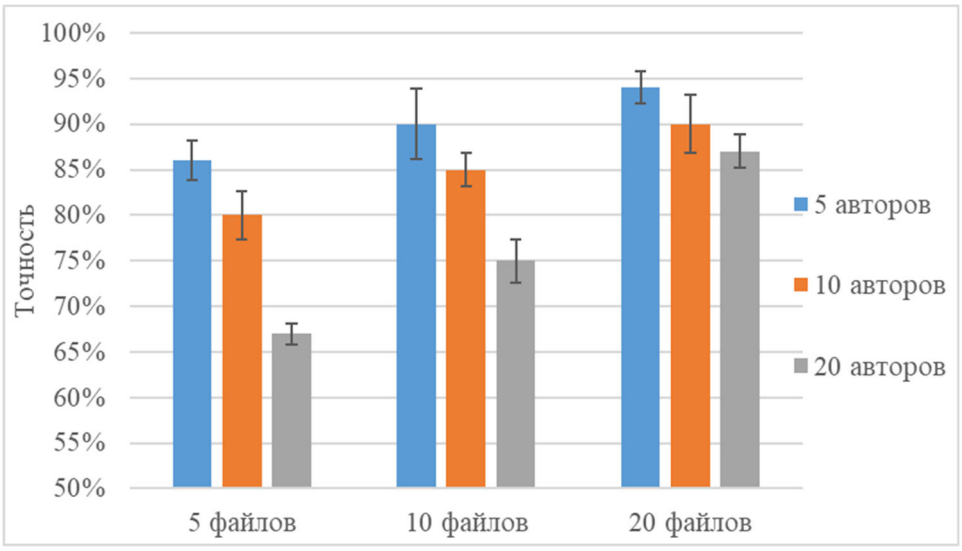


Рисунок 4.12 – Результаты экспериментов с коммитами

В таблице А.4 (Приложение А) приведено сравнение результатов ключевых работ по идентификации автора исходного кода с результатами, полученными в рамках данного исследования. Экспериментальные результаты были получены для набора данных, собранных с GitHub, и для открытого набора данных GCJ. Первый набор данных содержит коды разработчиков, программирующих на нескольких языках. Представленные в нем образцы зашумлены и не содержат явных особенностей стиля письма автора (это связано с соблюдением стандартов кодирования). Также они могут быть преобразованы сторонним ПО. Исходники одного и того же программиста могут содержать решения совершенно разных задач, что также может привести к запутыванию моделей классификации. Второй набор данных содержит не осложненные сторонними факторами образцы. Программы, исходные коды которых представлены в наборе, разработаны для решения однотипных задач, что позволяет модели фокусироваться только на авторских информативных признаках, не беря в учет различия в функциональности и назначении программ. За счет этого результаты, полученные на втором наборе данных, оказываются выше. Разница может достигать 10% в рамках одного и того же языка.

4.6 Выводы по главе

Предложенная методика показывает среднюю точность 93% в простых случаях идентификации автора исходного кода. Также она позволяет добиться высокой точности для исходных кодов, осложненных обфускацией и стандартами кодирования. Потеря точности в зависимости от используемого обфускатора в среднем составила 11%, что является признаком того, что методика устойчива к различным лексическим модификациями. Несмотря на то, что следование программиста стандартам кодирования оказывает негативное влияние на качество идентификации, методика позволяет достигнуть приемлемых для решения практических задач метрик качества путем увеличения объема обучающих данных и повысить точность с 30% до 80%.

Методика также продемонстрировала высокую точность для случая, когда программист владеет двумя и более языками. Точность для пары языков программирования достигает 92%, для 3 и более – 91%. Точность методики для идентификации автора, входящего в состав команды разработки, в зависимости от объема обучающих данных варьируется от 86% до 94%.

Полученные результаты позволяют сделать вывод о применимости разработанной методике к определению, в том числе, автора вредоносных программ.

Кроме того, методика оказалась не менее эффективной и для определения генеративной модели, сгенерировавшей исходный код программы. Средняя точность модели в задачах разграничения авторства между человеком-программистом и генеративной модели, составляет 94%.

Преимущества разработанной методики:

1. Высокая точность идентификации автора-программиста.
2. Устойчивость к обфускации и стандартам кодирования.
3. Независимость от языка, на котором написана программа, квалификации и опыта автора-программиста.
4. Высокая эффективность в случае командной разработки.

5. Способность к разграничению авторского и искусственно-сгенерированного кода, определению конкретной модели, создавшей искусственный код.

Для эффективного применения методики и получения качественных предсказаний необходимо следовать рекомендациям:

1. Рекомендуемый объем обучающей выборки составляет не менее 30 файлов на автора, число символов в каждом из которых составляет 1000 и более.

3. Автор каждого файла должен быть точно определен. В случае командной разработки необходимо разграничить коммиты авторов-программистов.

4. Рекомендуемые параметры CodeBERT: скорость обучения – 0,00003, коэффициент регуляризации – 0,015, размер батча – 16, число эпох обучения – 10.

5. Меньшее количество авторов-кандидатов и больший объем обучающих данных позволяют получить лучший результат идентификации.

5 ПРИМЕРЫ ПРИМЕНЕНИЯ РАЗРАБОТАННОЙ МЕТОДОЛОГИИ К РЕШЕНИЮ ВАЖНЫХ НАРОДНО-ХОЗЯЙСТВЕННЫХ ЗАДАЧ

В первом разделе подчеркнута важность решения задачи профилирования автора текста – определение пола и принадлежности к ЛГБТ-сообществу*, возрастной категории, настроения, взглядов автора текста и др. атрибутов для кибербезопасности. Актуальность исследования по данным темам связана как с их вспомогательной функцией при решении задач интеллектуального анализа текста, в частности, атрибуции (разд. 3), так и самостоятельного решения таких задач информационной безопасности, как:

− выявление лиц, приверженных к крайним, радикальным взглядам, позициям и мерам в общественной деятельности;

− дифференциация пользователей онлайн-платформ по возрасту для борьбы с педофилией и предотвращения доступа детей к контенту для взрослых;

− выявление лиц, принадлежащих к экстремистскому движению ЛГБТ*;

− определение плагиата.

5.1 Методика определения текстов деструктивной и экстремистской направленности

Методика предназначена для определения авторства русскоязычных комментариев пользователей социальных сетей, имеющих признаки экстремизма или деструктивную направленность [265, 293]. Это имеет важное значение для обнаружения потенциальных угроз и предотвращения терроризма и экстремизма.

* Запрещенная в Российской Федерации экстремистская организация.

5.1.1 Описание методики

Методика основывается на представленной в подразд. 3.2 базовой методике определения автора [20, 21, 193]. Проведена ее модернизация путем применения семантической сегментации, трансферного обучения и формирования информативного признакового пространства на их основе. Проведены эксперименты по детектированию деструктивного и экстремистского контента, а также использованию закрытой и открытой атрибуции.

Методика, учитывающая категорию деструктивных и экстремистских текстов представлена на рисунке 5.1.



Рисунок 5.1 – IDEF0 диаграмма определения текста деструктивной и экстремистской направленности

Специфика используемых данных (деструктивный и экстремистский контент) требует дополнительного этапа предварительной фильтрации сообщений социальных сетей и мессенджеров для формирования обучающего множества. Для учета данного фактора математическая постановка задачи определения автора из разд. 3 была модифицирована для использования тематических текстов.

Имеется эталонный датасет текстов $\mathbf{T}^e$, содержащий тексты интересующей тематики (в данном случае, деструктивной или экстремистской направленности). Также имеется целевой датасет $\mathbf{T}$, тексты которого имеют ту же тематику, однако содержат шумы в виде обычных текстов $\mathbf{T}'$. Необходимо отфильтровать тексты $\mathbf{T}''$ из множества $\mathbf{T}$ так, чтобы они не содержали текстов с шумом $\mathbf{T}'$:

$$\mathbf{T}'' = \mathbf{T} / \mathbf{T}'. \tag{5.1}$$

Для этого используется семантическая кластеризация, которая делит множество $\mathbf{T}^e$, на s подтем (кластеров):

$$clustering\left(\mathbf{T}^e, q\right) = \left\{\mathbf{T}_1^e, \mathbf{T}_2^e, ..., \mathbf{T}_s^e\right\}, \tag{5.2}$$

где q является кластеризационной метрикой.

Далее для каждого кластера извлекается список ключевых слов с помощью функции $keywords(\cdot)$ и отбирается n наиболее информативных из них с помощью функции $top(\cdot)$:

$$\mathbf{W}_i^e = top\left(keywords\left(\mathbf{T}_i^e\right), n\right), \tag{5.3}$$

Функции $keywords(\cdot)$ и $top(\cdot)$ определяются алгоритмом семантической кластеризации.

Итоговое множество ключевых слов формируется как объединение ключевых слов по всем подтемам:

$$\mathbf{W}^e = \bigcup_{i=1}^{s} \mathbf{W}_i^e. \tag{5.4}$$

Множество отфильтрованных текстов формируется как:

$$\mathbf{T}'' = \left\{ t_i \in \mathbf{T} \,|\, \left| \mathbf{W}^{t_i} \cap \mathbf{W}^{e} \right| > \theta \right\}, \tag{5.5}$$

где $\mathbf{W}^{t_i}$ – словарь текста t_i; θ – пороговое значение.

После фильтрации датасет $\mathbf{T}''$ содержит тексты интересующей тематики.

Эффективность решения задачи можно повысить за счет использования трансферного обучения и модели, способной решать смежную задачу. Такая модель уже содержит полезные знания. Ее использование совместно с отфильтрованными данными позволяет добиться лучших результатов при относительно небольшом обучающем датасете.

Пусть имеется смежная задача обучения $\mathrm{E}_s = \{\mathbf{Y}_s, P(\mathbf{Y}_s \,|\, \mathrm{X}_s)\}$ и домен $\Upsilon_s = \{\mathbf{X}_s \,|\, P(\mathrm{X}_s)\}$, где $\mathbf{X}_s$, $\mathbf{Y}_s$ – пространства признаков и меток. $\mathrm{X}_s = \{x_1, ..., x_s\} \in \mathbf{X}_s$ – частное распределение вероятностей. И есть интересующий домен данных $\Upsilon_t = \{\mathbf{X}_t \,|\, P(\mathrm{X}_s)\}$ и решаемая задача $\mathrm{E}_t = \{\mathbf{Y}_t, P(\mathbf{Y}_t \,|\, \mathrm{X}_t)\}$. Трансферное обучение направлено на улучшение обучения условного распределения вероятностей $P(Y_t \,|\, X_t)$ в Υ_t за счет информации, полученной из Υ_s и E_s, где $\mathrm{E}_t \neq \mathrm{E}_s$ или $\Upsilon_t \neq \Upsilon_s$.

В нашем случае можно использовать модель для определения экстремизма в тексте, предназначенную для других языков $(\mathbf{X}_t \neq \mathbf{X}_s)$, $(\mathbf{Y}_t = \mathbf{Y}_s)$. Либо модель, обученную на русскоязычных текстах для определения негативного окраса $(\mathbf{X}_t = \mathbf{X}_s)$, $(\mathbf{Y}_t \neq \mathbf{Y}_s)$.

5.1.2 Семантическая кластеризация

Семантическая кластеризация (подразд. 2.6) является полезным для целей определения авторства инструментом за счет ее способности к выделению связанных по смыслу кластеров слов и фраз. Анализ таких кластеров позволяет выявлять уникальные особенности, характерные для определенного автора или тематической группы текстов.

Для выявления наиболее эффективного подхода семантической кластеризации требуется ряд отдельных экспериментов, направленных на оценку соответствующих метрик качества [296].

Исходя из проведенного анализа современных трудов, посвященных задаче кластеризации текстов, было выделено несколько наиболее эффективных подходов к решению данной задачи. Среди них BERTopic, DBSCAN, KNN, NMF, LDA и агломеративная кластеризация, подробно описанные в подразд. 2.6.

Для оценки эффективности указанных методов кластеризации было использовано 3 метрики, не зависящие от реальных меток.

Коэффициент силуэта (silhouette_score) [198] определяет расстояние от объекта до ближайшего кластера, в состав которого он не входит. Чем ближе значение коэффициента силуэта к 1, тем выше качество кластеризации. На полностью неверную кластеризацию указывает значение равное –1. Метрика рассчитывается как:

$$silhouette\_score = \frac{\sum_{i=1}^{n} silhouette\_score(i)}{n}, \tag{5.6}$$

$$silhouette\_score(i) = \frac{b_i - a_i}{max(a_i, b_i)}, \tag{5.7}$$

где a_i – среднее расстояние от объекта *i* до объектов, образующих тот же класс; b_i – среднее расстояние *i*-го объекта до всех объектов в соседнем кластере; *n* – количество кластеров.

Индекс Дэвиса – Болдина (db_score) [60] оценивает сходство наиболее близких друг к другу кластеров. При лучшей кластеризации значение индекса будет близким к 1, при низком качестве кластеризации к 0. Индекс Дэвиса – Болдина считается как:

$$db\_score = \frac{1}{n}\sum_{i=1}^{n} R_i, \tag{5.8}$$

$$R_i = \max\left(R_{ij}\right), i = 1,\ldots,n_c, \tag{5.9}$$

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}, \tag{5.10}$$

где s_i – разброс внутри кластера *i*, который должен быть как можно меньше; s_j – разброс внутри кластера *j*, который должен быть как можно меньше; d_{ij} – расстояние между *i*-м и *j*-м кластером, которое должен быть как можно больше.

Индекс Калински – Харабаза (*ch_score*) [51] используется для оценки разницы между кластерами и их компактности относительно друг друга. Чем более разделимы и компактны кластеры, тем выше значение индекса. Рассчитывается как:

$$ch\_score = \frac{BGSS}{WGSS} \cdot \frac{N-K}{K-1}, \tag{5.11}$$

где N – общее число наблюдений; *K* – общее число кластеров; *BGSS* – межгрупповая дисперсия; *WGSS* – внутригрупповая дисперсия.

$$WGSS = \sum_{k=1}^{K} \| X_{i_k} - C_k \|^2 , \tag{5.12}$$

где n_k – количество наблюдений в кластере *k*; X_{i_k} – *i*-е наблюдение кластера *k*; C_k – центр тяжести кластера *k*.

$$BGSS = \sum_{k=1}^{K} n_k \cdot \| C_k - C \|^2 , \tag{5.13}$$

где *C* – барицентр датасета.

Для экспериментов использовались наборы данных, описанные в п. 6.7.4. Результаты эксперимента приведены в таблице 5.1.

Важно отметить, что при выборе метода семантической кластеризации в рамках данного исследования более важную роль играют коэффициент силуэта и индекс Дэвиса – Болдина. Это обусловлено тем, что они оценивают качество семантической кластеризации на основе наблюдений внутри каждого класса, а не между классами в целом. Таким образом, был сделан вывод о том, что BERTopic является наиболее эффективным для решения поставленной задачи и может быть использован в качестве основы методики.

Таблица 5.1 – Результаты экспериментов

Набор данных	Метод	silhouette_score	db_score	ch_score
Kaggle	BERTopic	**0,728**	**0,59**	48,122
	DBSCAN	0,714	1,011	41,475
	KNN	0,021	6,13	81,351
	LDA	-0,0003	28,634	45,711
	NMF	0,008	6,608	**236,726**
	Агломеративный	0,491	3,891	59,327
KavkazChat*	BERTopic	**0,584**	**0,602**	**138,871**
	DBSCAN	0,122	1,445	2,382
	KNN	0,064	2,95	2,192
	LDA	0,004	11,678	1,881
	NMF	0,006	8,725	3,88
	Агломеративный	0,18	0,972	2,742
Иностранные агенты	BERTopic	**0,557**	0,66	**134,36**
	DBSCAN	0,441	10,341	2,67
	KNN	0,412	**0,49**	2,952
	LDA	0,429	10,875	2,57
	NMF	0,441	9,41	3,06
	Агломеративный	0,54	0,794	10,38

На рисунке 5.2 представлена техника оптимизации параметров BERTopic на каждом из этапов его работы.

Первый этап работы BERTopic состоит в формировании векторных представлений, полученных на входе текстов. Такие представления формируются при помощи SentenceTransformer [194]. Данная модель основана на трансформерах и преобразует предложения в векторные представления. Она обучается на больших корпусах текстов и учитывает семантику предложений при вычислении этих представлений. За счет применения SentenceTransformer, BERTopic обеспечивает высокую семантическую близость между предложениями.

Следующий этап направлен на семантическую кластеризацию. Векторные представления, полученные SentenceTransformer, передаются в алгоритм Uniform Manifold Approximation and Projection (UMAP) [207] для снижения размерности без потери структуры данных. Применение UMAP оказывает положительное влияние

* Признан экстремистским в Российской Федерации.

на работу алгоритмов кластеризации и позволяет оптимизировать процесс. Непосредственно процесс кластеризации [98] может выполняться с помощью различных алгоритмов, встроенных в BERTopic. Среди них иерархический DBSCAN (HDBSCAN), KNN, Birch и другие известные алгоритмы.

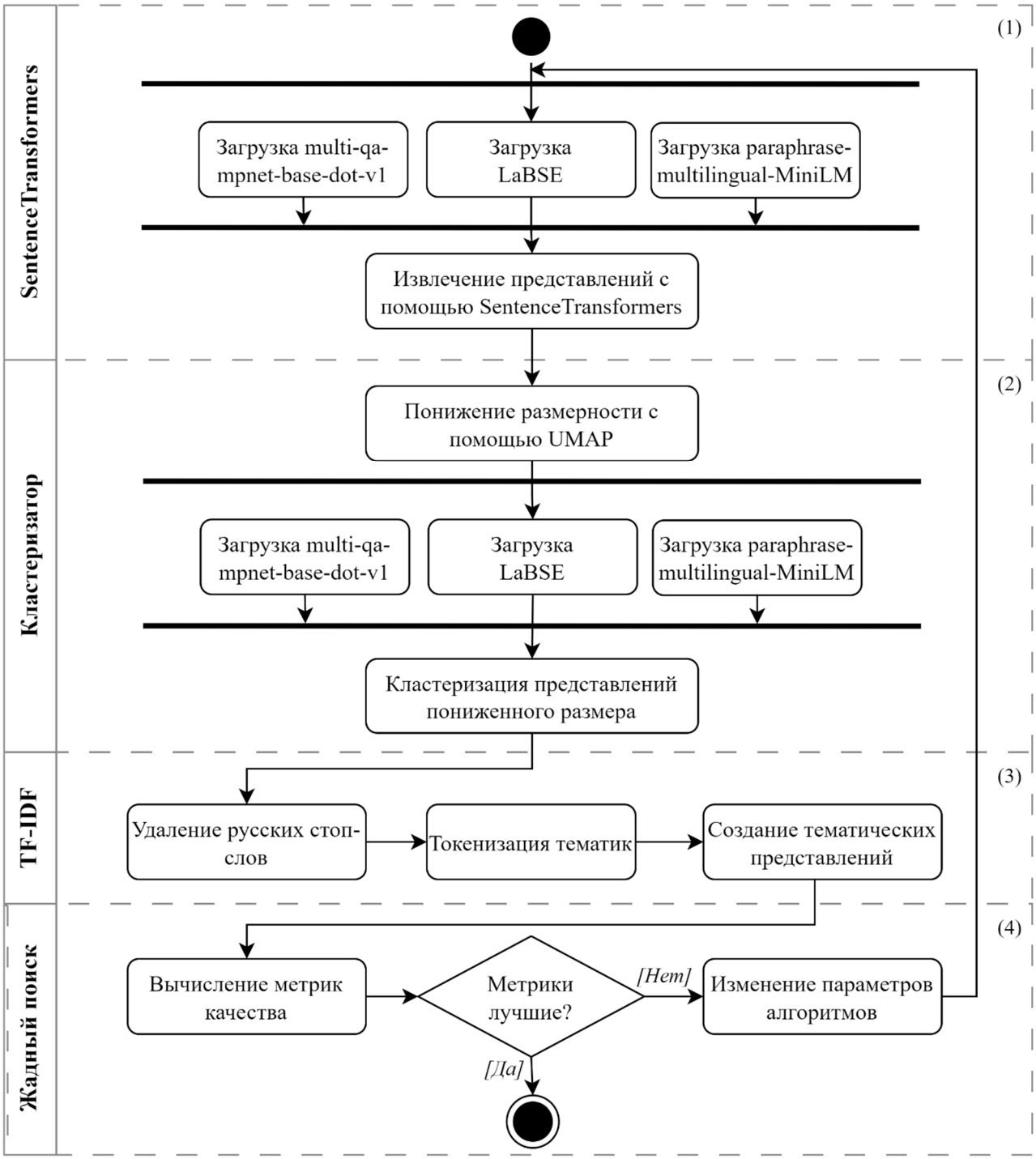


Рисунок 5.2 – Семантическая кластеризация при помощи BERTopic

Третий этап направлен на выбор представлений, соответствующих определенной теме, и их ранжирование. С этой целью используются методы TF-IDF и метод максимальной предельной релевантности (MMR).

В рамках исследования было опробовано несколько сочетаний алгоритмов BERTopic. В качестве моделей, используемых SentenceTransformer для построения представлений, были выбраны multi-qa-mpnet-base-dot-v1 (M1), paraphrase-multilingual-MiniLM-L12-v2 (M2) и BERT (LaBSE) (M3) [127]. В качестве внутренних алгоритмов кластеризации: KNN по методу локтя, Birch и HDBSCAN. Результаты подборки параметров жадным поиском представлены в таблице 5.2.

Таблица 5.2 – Результаты экспериментов с BERTopic

Набор данных	Метрика	Sentence Transformer	Алгоритм кластеризации		
			KNN	Birch	HDBSCAN
Kaggle	Коэффициент силуэта	M1	0,221	0,424	0,593
		M2	0,363	0,552	0,742
		M3	0,332	0,543	**0,757**
	Индекс Дэвиса – Болдина	M1	0,714	0,581	0,69
		M2	0,662	0,491	**0,321**
		M3	0,555	0,432	**0,321**
	Индекс Калински – Харабаза	M1	13,23	22,183	110,43
		M2	27,452	49,115	116,213
		M3	38,335	59,654	**127,35**
KavkazChat*	Коэффициент силуэта	M1	0,103	0,242	0,624
		M2	0,31	0,455	0,632
		M3	0,244	0,391	**0,643**
	Индекс Дэвиса – Болдина	M1	0,613	0,563	0,491
		M2	0,592	0,472	0,454
		M3	0,575	0,472	**0,442**
	Индекс Калински – Харабаза	M1	10,561	104,61	112,91
		M2	68,174	109,03	136,254
		M3	75,153	107,27	**144,53**
Иностранные агенты	Коэффициент силуэта	M1	0,483	0,525	0,593
		M2	0,494	0,62	0,647
		M3	0,532	0,615	**0,654**

* Признан экстремистским в Российской Федерации.

Набор данных	Метрика	Sentence Transformer	Алгоритм кластеризации		
			KNN	Birch	HDBSCAN
	Индекс Дэвиса – Болдина	M1	0,686	0,519	0,381
		M2	0,62	0,487	**0,321**
		M3	0,553	0,431	**0,321**
	Индекс Калински – Харабаза	M1	85,59	116,69	115,1
		M2	91,54	119,23	116,12
		M3	92,35	128,13	**137,64**

Согласно полученным результатам, в качестве алгоритма кластеризации был выбран HDBSCAN по метрике Евклида, в качестве алгоритма представления текстов – LaBSE, в качестве алгоритма понижения размерности – UMAP по метрике Жаккара. Такая конфигурация BERTopic позволила добиться показателей 0,757 по коэффициенту силуэта, 0,321 по индексу Дэвиса – Болдина и 127,5 по индексу Калински – Харабаза. Полученные результаты указывают на то, что кластеры не перекрываются и имеют достаточный уровень разделения.

Параметры, подобранные для UMAP при помощи жадного поиска: количество соседей – 15, количество компонентов – 3, метрика – Жаккар. Для HDBSCAN: метод отбора – eom, метрика – Евклид.

Визуализация кластеров UMAP, полученных BERTopic показана на рисунке 5.3, а также приведены примеры облака слов, соответствующие топ-5 слов для выделенных кластеров.

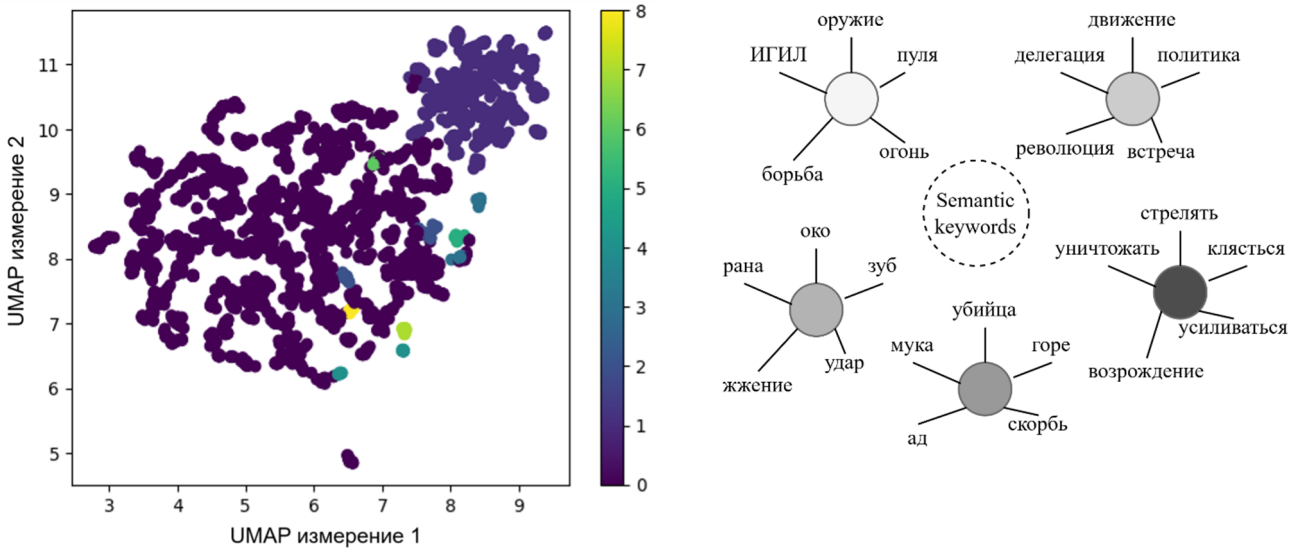


Рисунок 5.3 – Визуализация UMAP и примеры ключевых слов топ-5 кластеров

Разработанная методика семантической кластеризации может быть особенно полезна в задачах анализа и классификации больших объемов текстовых данных. Формирование кластеров, содержащих тексты, относящиеся к конкретным темам или категориям согласно их семантическому смыслу, таким как негативный эмоциональный окрас, токсичность, экстремизм, терроризм и т.д., позволит более эффективно выявлять и анализировать данные для выявления потенциально вредоносной и противозаконной информации.

5.1.3 Трансферное обучение

На этапе обучения модели применяется техника трансферного обучения, использующая модель, обученную решать смежную задачу.

При поиске подходящей для переноса модели ограничивающими факторами были поддержка моделью русского языка и ее способность к выявлению негативной тональности. Критериям удовлетворяют Russian Toxicity Classifier [190], Emotion Detection [72] и Russian Sensitive Topics [189].

Среди представленных вариантов была выбрана модель Russian sensitive topics [189], описанная авторами в статье [25], поскольку помимо стандартного разделения по эмоциональной окраске текста, авторами была проведена более глубокая работа, связанная с выделением потенциально опасных для физического и ментального здоровья тематик в тексте, например, убийства, пропаганда наркотиков и педофилии, проституция, радикальные политические взгляды, травля. Описания таких действий характерны для запрещенных законом текстов. Модель обучена на размеченном экспертами вручную наборе данных, состоящем из 82000 русскоязычных сообщений. При обучении модели авторы настраивали предварительно обученную модель ruBERT 2019 на собственных данных. Модель использовалась в реализации DeepPavlov.

5.1.4 Определение автора текста, имеющего деструктивную и экстремистскую направленность

Помимо определения факта наличия признаков экстремизма в тексте или деструктивной направленности можно также определять и автора такого текста. При этом учитывать случай закрытого и открытого множества потенциальных авторов. Используемые методики описаны в подразд. 3.2 и 3.5.

5.1.5 Результаты исследования методики

Пп. 5.1.5.1 содержит результаты оценки качества данных, включая верификацию распространителей деструктивных и экстремистских текстов, пп. 5.1.5.2 – результаты экспериментов по определению деструктивного и экстремистского контента в тексте, пп. 5.1.5.3 – определения авторства текста в случае закрытого множества кандидатов, пп. 5.1.5.4. – определение авторства в случае открытого множества кандидатов. В пп. 5.1.5.5 приведены результаты оценки эффективности применения семантической кластеризации и трансферного обучения в сравнении с разработанной в главе 3 методикой. В пп. 5.1.5.6 методика сравнивается с работами других исследователей, в пп. 5.1.5.7 с современными генеративными моделями.

5.1.5.1 Оценка качества данных

Для проверки качества данных были проведены первичные проверки на отсутствие дубликатов и пустых значений. Затем для проверки однородности данных использовался критерий Хи-квадрат. Нулевая гипотеза (*H0*) – распределение данных однородно (нет существенной разницы). Альтернативная гипотеза (*H1*) – распределение данных неоднородно (значимая разница). Результаты проведения

теста для каждого датасета в отдельности и их объединений приведены в таблице 5.3.

Таблица 5.3 – Результаты теста по проверке однородности данных

	ВКонтакте	ВКонтакте сгенерированные	Telegram экстремисты	Telegram иноагенты	Telegram новости	KavkazChat*
ВКонтакте	**0,31**	**0,37**	0,0007	0,0008	0,001	0,02
ВКонтакте сгенерированные	**0,37**	**0,45**	0,004	0,006	0,03	0,01
Telegram экстремисты	0,0007	0,004	**0,58**	**0,38**	**0,33**	0,02
Telegram иноагенты	0,0008	0,006	**0,38**	**0,44**	**0,19**	0,0006
Telegram новости	0,001	0,03	**0,33**	**0,19**	**1,24**	0,04
KavkazChat*	0,02	0,01	0,02	0,0006	0,04	**0,23**

В таблице 5.3 жирным шрифтом выделены значения *p* больше, чем уровень значимости 0,05. Таким образом, поскольку *p*-значения 0,31, 0,45, 0,58, 1,24. 0,23 для каждого из датасетов в отдельности превышают уровень значимости (0,05), принимается нулевая гипотеза – распределение данных однородно. Также нулевая гипотеза принимается для случая объединения датасетов ВКонтакте и ВКонтакте сгенерированные, Telegram экстремисты – иноагенты и Telegram новости (*p*-значения 0,37 и 0,33 соответственно), что свидетельствует об однородности данных в объединенных датасетах. Во всех остальных случаях нулевая гипотеза отвергается и принимается альтернативная о неоднородности данных, а значит наборы данных не следует объединять при проведении экспериментов в единый датасет.

* Признан экстремистским в Российской Федерации.

5.1.5.2 Определение и верификация текстов, содержащих признаки экстремизма или деструктивной направленности

При определении наличия в тексте контента, связанного с экстремизмом или деструктивных текстов решается задача бинарной классификации. Использовались датасеты Telegram экстремистов и иноагентов, а также KavkazChat*. Датасет комментариев ВКонтакте не был использован, так как это могло оказать негативное влияние на однородность данных. Деструктивные тексты составили первый класс. Второй класс содержал тексты авторов, пишущих на обычные темы (Telegram новости в случае датасетов из Telegram или тексты на бытовые темы из набора KavkazChat*). Использовалось от 15000 до 30000 образцов для каждого класса в зависимости от датасета. Все используемые датасеты были сбалансированы. Результаты представлены в таблице 5.4.

Таблица 5.4 – Результаты определения деструктивного и экстремистского контента в тексте

Датасет	Число образцов в классе	Точность, %
Telegram экстремисты	15000	94±5
KavkazChat*	30000	86±6
Общий датасет экстремистов	59000	90±4
Telegram иноагенты	20000	90±8

Согласно результатам, точность определения контента, созданного экстремистами и иностранными агентами, составляет более 90% с условием наличия 15000 и более образцов каждого класса. В случае KavkazChat* точность составила 86% при 30000 образцах каждого класса. В целом результаты, полученные на датасете KavkazChat* ниже, чем при проведении аналогичных экспериментов для датасетов из Telegram, что подтверждает результаты проверки однородности данных и ее значимости при обучении модели.

Альтернативным методом является использование механизма верификации с помощью One-Class SVM (см. подразд. 3.4), обученного только на текстах,

содержащих деструктивные тексты. В процессе тестирования подавались образцы, не задействованные для обучения и тексты, созданные организациями и физическими лицами, не связанными с противозаконной деятельностью. Также проведена оценка точности в зависимости от выбора векторного представления текста – в виде эмбеддингов, полученных на основе трансферной модели или в виде распределения частот ключевых слов, полученных на этапе семантической кластеризации. Результаты верификации приведены в таблице 5.5.

Таблица 5.5 – Результаты верификации распространителей деструктивных и экстремистских текстов

Набор данных	Представление текста	Точность, %
Telegram экстремисты	Эмбеддинги	88±3%
	Частоты ключевых слов	77±3%
KavkazChat*	Эмбеддинги	81±2%
	Частоты ключевых слов	62±3%
Общий датасет экстремистов, включая переведенные датасеты	Эмбеддинги	90±3%
	Частоты ключевых слов	77±4%
Telegram иноагенты	Эмбеддинги	79±3%
	Частоты ключевых слов	70±4%

Для всех четырех наборов данных использование эмбеддингов обеспечило прирост точности верификации (до 19,3%). Максимальная точность верификации достигнута для общего набора экстремистских текстов – 90%.

Обобщая результаты обоих экспериментов, направленных на детектирование экстремистского контента, отметим, что точность бинарного классификатора выше, чем одноклассового на 4%. Поэтому использование обучающего набора, включающего тексты двух классов (деструктивные и «нормальные»), является предпочтительным.

* Признан экстремистским в Российской Федерации.

5.1.5.3 Определение авторства текстов деструктивной и экстремистской направленности (закрытое множество кандидатов)

Целью являлось сравнение результатов, полученных на датасетах из разных источников (Telegram, KavkazChat*), а также оценка возможности переноса знаний предобученной на негативных текстах модели для поиска комментариев, содержащих деструктивную направленность, и повышения за счет этого точности определения автора.

Эксперименты по идентификации автора для закрытого множества кандидатов проводились для 2, 5, 10 и 20 авторов каждого набора данных (таблица 5.6). Количество образцов каждого класса для обучения составляло 40, для тестирования 10.

Таблица 5.6 – Результаты определения автора в случае закрытой атрибуции

Кол-во авторов	Точность, %			
	Telegram экстремисты	KavkazChat*	Telegram иноагенты	Telegram новости
2	94±4	87±4	95±3	94±5
5	90±3	80±9	91±3	85±6
10	80±4	72±6	77±4	78±8
20	71±6	54±9	67±5	54±1

Как и в предыдущих случаях результаты на датасетах из Telegram выше, чем на KavkazChat*. При классификации 2 авторов лучшая точность 95% достигается для иноагентов, при пяти авторах – 91%. Это свидетельствует о способности модели к определению авторов внутри группы комментаторов сообществ с близкой тематикой. Усложнение задачи приводит к закономерному снижению точности во всех случаях. Моделям становится сложно различать авторов из-за схожих паттернов в авторском стиле. Потому что использованная для трансферного обучения модель была создана для определения негативно окрашенных текстов, а в случае деструктивных сообществ большинство комментариев являются именно такими.

* Признан экстремистским в Российской Федерации.

5.1.5.4 Определение авторства текстов деструктивной и экстремистской направленности (открытое множество кандидатов)

Цель эксперимента состояла в определении был ли создан анонимный текст одним из известных авторов деструктивного или экстремистского контента или это автор, которого не было обучающем множестве (null-класс). Методика открытой атрибуции соответствует представленной в подразд. 3.5. В качестве признаков для обучения применялись эмбеддинги, полученные на основе модели BERT для трансферного обучения. Результаты представлены в таблице 5.7.

Таблица 5.7 – Результаты определения автора в случае открытой атрибуции

Кол-во авторов	Точность, %			
	Telegram экстремисты	KavkazChat*	Общий датасет экстремистов	Telegram иноагенты
2+null	93,5±2,4	87±4	90±2	92±2
4+null	90,1±3,6	80±3	85±4	87±3
9+null	78,5±4,2	70±3,5	78±3	76±3
19+null	70,4±1,9	52±4	68±3	61±2

В случае открытой атрибуции модель способна определять автора с точностью до 93%. В целом, точность для случая открытого множества кандидатов ниже на 1–3% по всем наборам. Точность идентификации автора для датасета KavkazChat* ниже, как и было в случае закрытого множества кандидатов. Дополнительно был проведен эксперимент по объединению всех датасетов, содержащих экстремистские тексты. Точность для такого датасета ожидаемо соответствует среднему арифметическому значению. Также можно отметить, что тексты экстремистов оказались более простыми для идентификации, чем тексты иностранных агентов. Разница в точности при равном количестве классов составляет 2–10%.

* Признан экстремистским в Российской Федерации.

5.1.5.5 Оценка разработанной методики в сравнении с базовой методикой определения автора текста

В разделе продемонстрированы преимущества использования семантической кластеризации и трансферного обучения путем их последовательного добавления к базовой методике (см. разд. 3) и оценке результатов (рисунки 5.4–5.6).

Из результатов видно, что добавление трансферного обучения позволяет получить прирост точности до 10%, 7% и 12% в случае закрытого и открытого множеств авторов и определения деструктивного контента, соответственно. Добавление семантической кластеризации – 9%, 5% и 9% для тех же случаев. Совместное использование двух инструментов дает совокупный прирост точности до 12%, 14% и 20% для обоих случаев определения авторства и деструктивного контента.

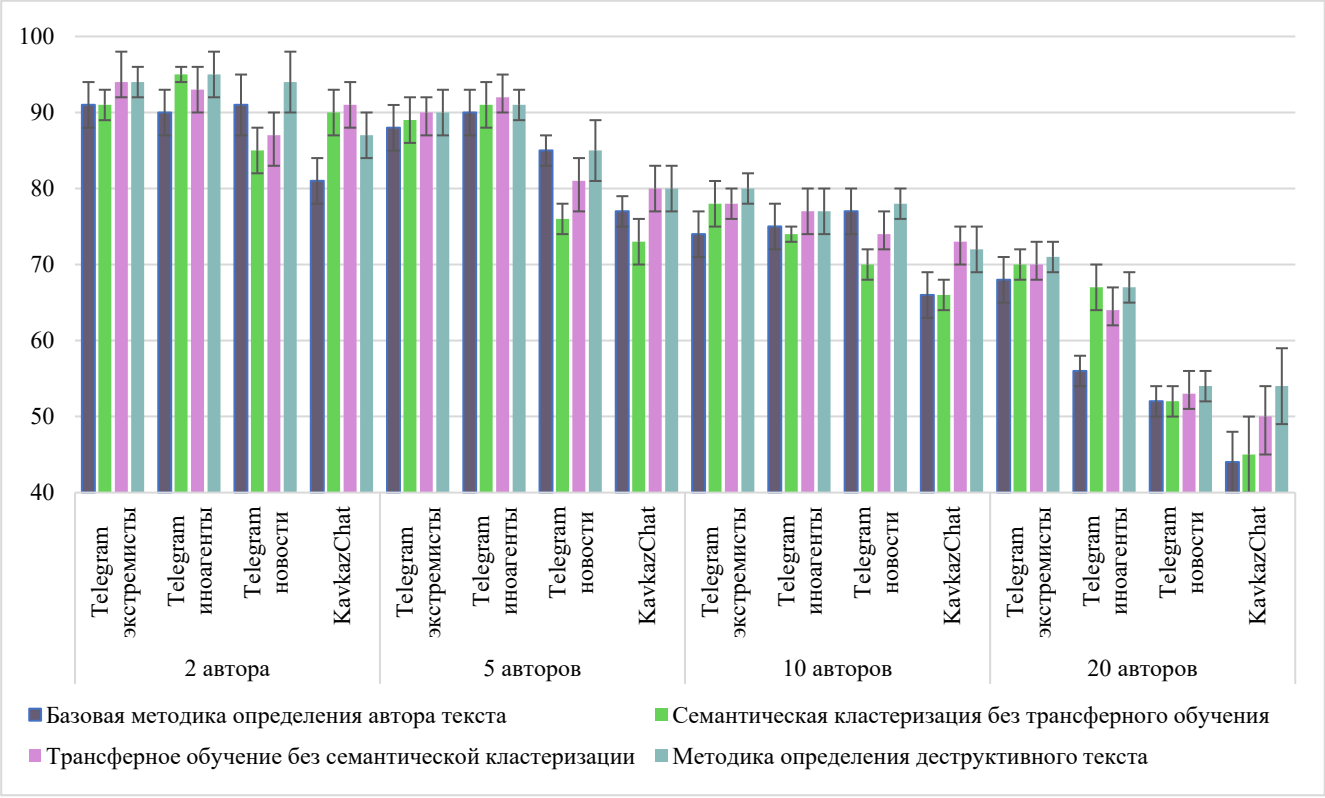


Рисунок 5.4 – Сравнение методик закрытой атрибуции

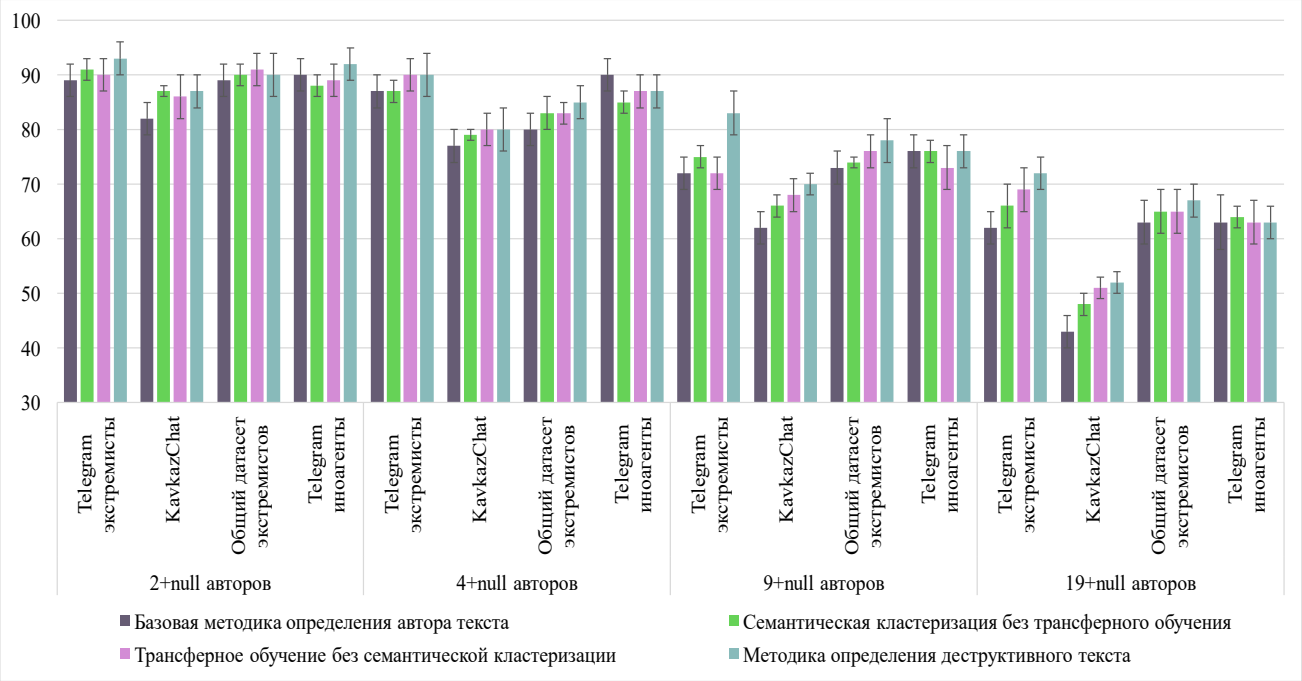


Рисунок 5.5 – Сравнение методик открытой атрибуции

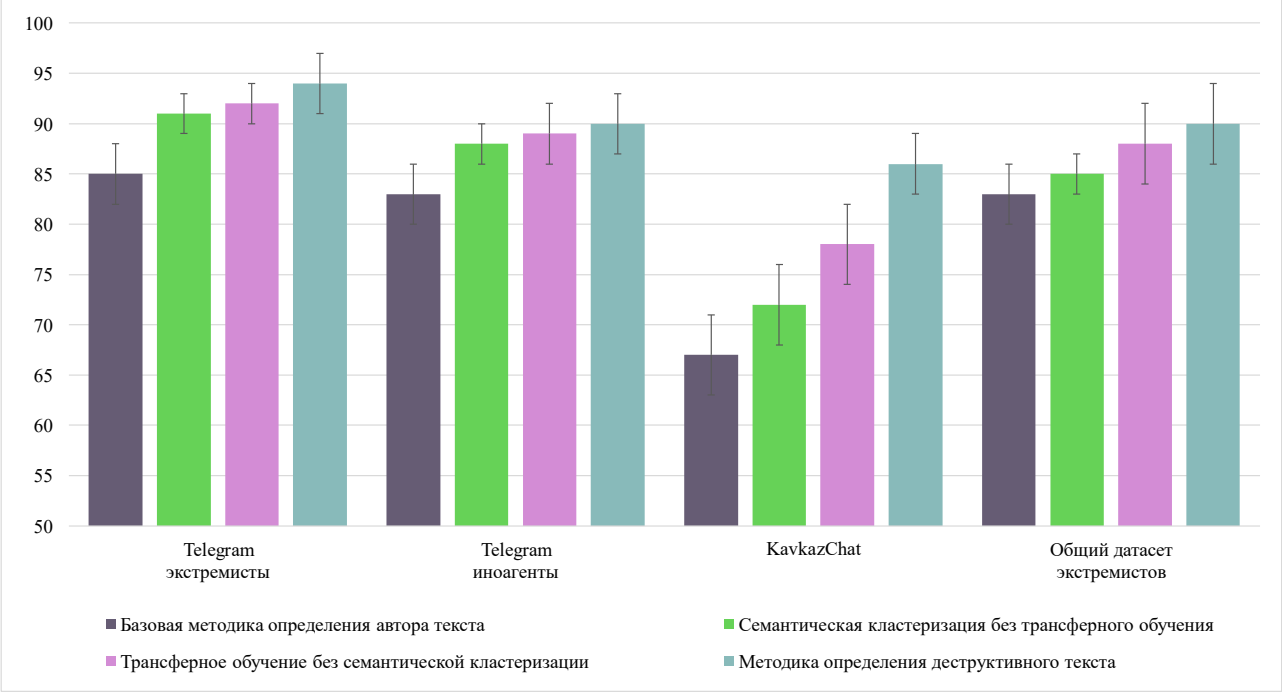


Рисунок 5.6 – Сравнение методик определения деструктивного и экстремистского контента

5.1.5.6 Оценка разработанной методики в сравнении с аналогами

Наиболее важным аспектом при оценке полученной методики являются ее эффективность, преимущества и ограничения в сравнении с аналогами для закрытого множества кандидатов и деструктивного контента (Приложение А, таблица А.5, раздел «Определение деструктивной и экстремистской направленности в тексте»). Полученные значения точности для таких случаев (2 и 5 авторов) на 9–13% превышают результаты, представленные другими исследователями. В случае 10 и 20 авторов прирост точности в сравнении с аналогами возрастает до 22%. Наиболее объективным является сравнение с работой Т.А. Литвиновой и соавторов [135], где использовали тот же датасет KavkazChat* для экспериментов с 20 авторами. Более высокая точность (55% против 49%) подтверждает эффективность семантической кластеризации и трансферного обучения в сравнении с классическими подходами на основе ручного выбора лемм для фильтрации датасетов.

Разработанная модель определения экстремизма обучена на 30000, 60000 и 118000 образцах на наборах данных Telegram, KavkazChat* и общего экстремистского датасета, включая переведенные тексты (15000, 30000 и 59000 в классе для перечисленных наборов, соответственно). Точность определения деструктивного контента – 94%. Аналоги уступают в точности от 1 до 32%, обучены на меньших наборах данных и не используют методы фильтрации для отсеивания недеструктивных образцов.

* Признан экстремистским в Российской Федерации.

5.1.5.7 Оценка разработанной методики в сравнении с современными генеративными моделями

Чтобы оценить практическую значимость предложенной методики, было решено провести сравнение современных моделей ИИ: ChatGPT 4 [46], ChatGPT 4-o [47], YandexGPT 2 [220] и YandexGPT 3 [221] для обнаружения текстов экстремистской направленности. Эти генеративные модели используются для решения широкого спектра задач в области обработки естественного язык. Модели от компании «Яндекс» специально разработаны для анализа текстов на русском языке.

Сравнение моделей для задачи определения авторства не проводилось, поскольку определение автора текста строго зависит от обучающих данных. Взаимодействие с моделями осуществляется посредством запросов ограниченной длины. Подать нужное количество обучающих текстов для 5 и более авторов не представляется возможным из-за ограниченного контекстного окна моделей.

В ходе экспериментов было использовано 1000 текстов, половина из которых содержала тексты, содержащие экстремистские высказывания. Соблюдались все рекомендуемые правила составления промптов. Для каждой из моделей они были составлены идентично, за исключением языковых различий: для YandexGPT он был на русском языке, а для моделей ChatGPT четвертой версии – на английском.

Модели YandexGPT 2 и 3 при подаче на вход строки, содержащей чувствительные темы, не давали однозначного ответа. В большинстве случаев полученный результат был «К сожалению, я не могу ничего сказать об этом. Давайте сменим тему?». Попытки конкретизации инструкций при составлении запроса для получения однозначного ответа «Да/Нет» не принесли результата. В результате оценить точность моделей Yandex GPT в этом эксперименте не представляется возможным. Результаты представлены в таблице 5.8.

Таблица 5.8 – Сравнение современных моделей ИИ в задаче обнаружения текста с признаками экстремизма

Модель	Точность определения экстремистских текстов, %	Точность определения не экстремистских текстов, %	Общая точность решения задачи, %
Романов А.С.	90%	87%	89%
ChatGPT 4	80%	66%	73%
ChatGPT 4-o	82%	61%	72%
Yandex GPT 2	н/д (25 ответов из 500 запросов)	н/д (20 ответов из 500 запросов)	н/д
Yandex GPT 3	н/д (40 ответов из 500 запросов)	82% (410 ответов из 500 запросов)	н/д

Согласно полученным результатам, доля правильных ответов модели, представленной в диссертационной работе, составила 89%, что соответствует полученным ранее результатам (см. таблицу 5.5). Поскольку задача связана с классификацией русскоязычных текстов, ChatGPT, приспособленный для решения задач обработки естественного языка на английском языке, работает хуже, чем специально обученная для этой задачи модель.

5.1.6 Выводы по подразделу

Общие рекомендации по использованию методики включают:

1. В случае необходимости фильтрации датасета, содержащего деструктивный контент, на основе семантической кластеризации, следует иметь эталонный датасет, где заведомо известно, что все образцы текстов содержат деструктивный (экстремистский) контент.

2. При выборе исходной модели для процесса трансферного обучения следует использовать предварительно обученные модели, предназначенные для анализа русскоязычных текстов или разработать собственную модель.

3. Семантическую кластеризацию следует использовать со следующими параметрами: UMAP: число соседей – 15; количество измерений – 3, метрика – Жаккара; HDBSCAN: метрика – расстояние Евклида, метод отбора кластеров – eom.

4. Следует использовать следующие параметры GRU+CNN: embedding слой, размерность – 200, сверточный слой: размерность – 4096, размер ядра свертки – 3, функция активации – ReLU, dropout слой, коэффициент – 0,2, GlobalMaxPooling, преобразование выходного массива в трехмерную форму для подачи на GRU-слой, GRU слой: размерность – 128, кол-во нейронов в выходном слое – 2 (деструктивный или недеструктивный текст) или количеству авторов – 2, 5, 10 или 20; функция активации – сигмоидальная (для определения текстов деструктивной направленности) или SoftMax (определение автора).

5. Параметры One-Class SVM: алгоритм обучения – sequential optimization method, ядро – линейное, параметр регуляризации – 0,1, допустимый уровень ошибки – 0,00001, нормирование – включено.

К ограничениям можно отнести:

1. Процесс трансферного обучения, нацеленный на выявление материалов, пропагандирующих потенциально опасные идеи, может быть зависим от контекста и культурных особенностей, что усложняет разработку универсальной методики.

2. Показано, что семантическая кластеризация является важным этапом методики для формирования обучающего набора. При правильном отборе тематических текстов модель способна точно определять класс тестового образца.

3. При формировании набора данных следует принять во внимание шумы и избыточную информацию, публикуемую пользователями, а также рекламный контент, не связанный с тематикой сообщества или канала.

4. Совместное использование семантической кластеризации и трансферного обучения является сложным процессом, потому что требуется как минимум две модели. Первая – содержащая знания смежной предметной области, вторая – целевая для решаемой задачи. В сравнении с нейронными сетями и классическими методами машинного обучения, этот процесс требует более тщательной настройки, больших вычислительных ресурсов и экспериментов для подбора параметров.

5.2 Методика определения возраста автора текстовой информации

Данный раздел посвящен определению возраста автора текста [64, 187, 266]. Важность исследования состоит в разработке методики, нацеленной на дифференциацию авторов текстов, публикуемых в сети, по возрастным группам.

5.2.1 Описание методики

Методика представлена на рисунке 5.7.

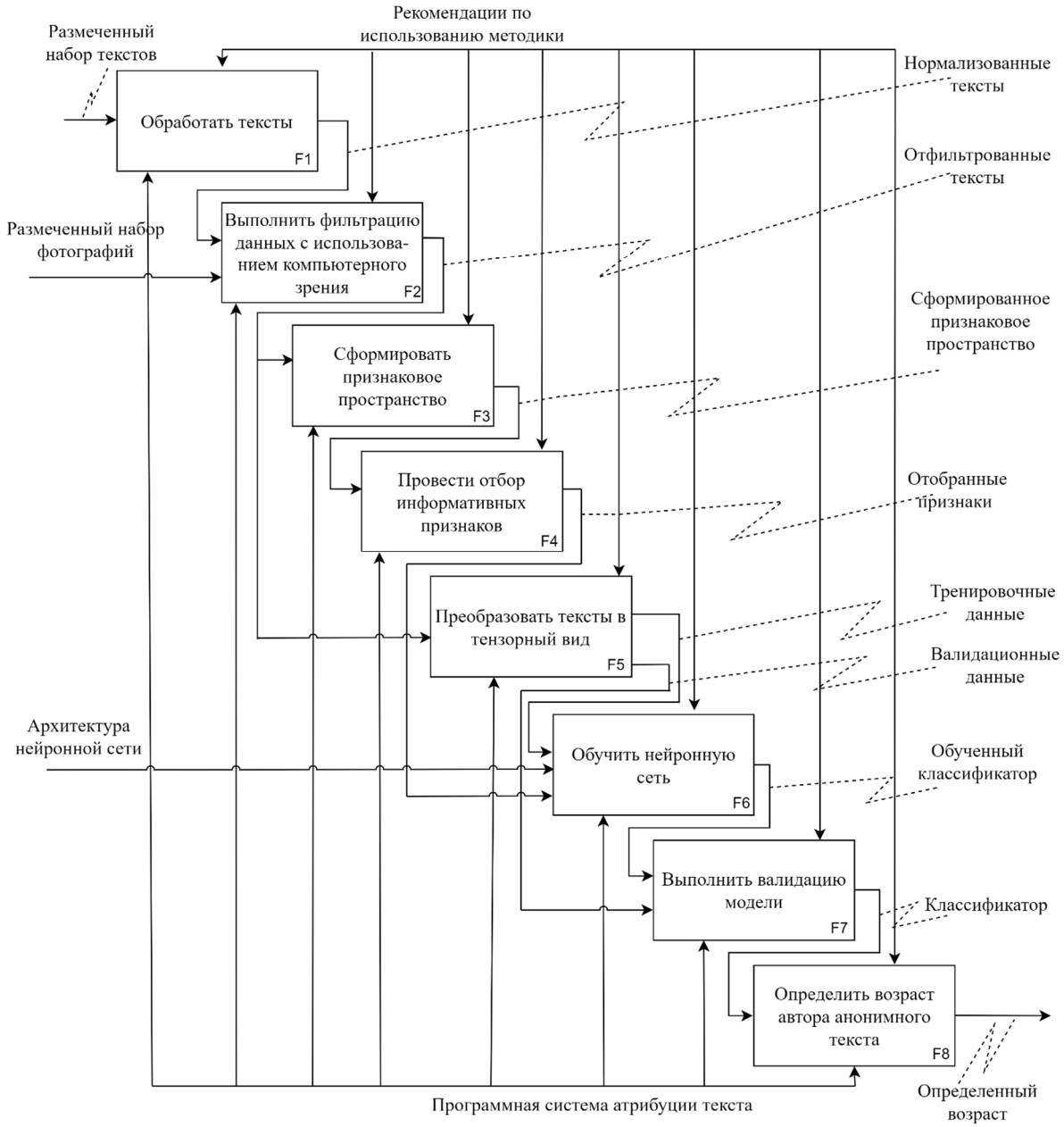


Рисунок 5.7 – Обобщенная методика определения возраста автора

5.2.1.1 Предварительная обработка текстов

Этап предварительной обработки предполагает очистку текстов от зашумленных данных. Зашумленными данными в задаче определения автора и его характеристик являются спам-сообщения, публикуемые пользователями социальных сетей. Фильтрация таких сообщений состоит в удалении текстов, содержащих слова: «актив», «подписаться», «лайк», «взломать», «взаимная подписка» и т.д.

Специфика публикуемых на онлайн-платформах текстов состоит в использовании комментариев и сообщений, содержащих преимущественно эмоджи. Исходя из этого, также удаляются тексты, состоящие в основном из смайлов и включающие менее 5 русских слов. Все смайлы заменяются тегом «@emoji».

Последним шагом предобработки текстов является их лемматизация и стемминг с целью приведения слов к начальной форме для упрощения понимания моделью классификации.

5.2.2.2 Фильтрация недостоверных данных о возрасте

Часто пользователи соцсетей указывают в профиле неправильный возраст. Несовершеннолетние делают это с целью получения доступа к контенту для взрослых или прямой регистрации на онлайн-платформе. Мотивация взрослых людей различна и, как правило, не несет потенциальных угроз. Однако использование таких текстов для обучения модели может привести к неверным результатам.

Для решения этой проблемы было решено использовать фотографии со страниц пользователей и модель компьютерного зрения как инструмент эффективной фильтрации недостоверных данных.

Большинство подходов к решению задачи определения возраста автора основано на CNN архитектурах и их более сложных модификациях. В процессе разработки методики было установлено [64], что VGG-Face является наиболее эффективной моделью для определения возраста пользователя по его фото. Фильтрация

недостоверных данных с помощью VGG-Face включает несколько шагов. Сначала модель определяет возраст. Затем к предсказанному моделью возрасту прибавляется или вычитается 2 года. В датасет добавляются только тексты тех пользователей, чей указанный возраст совпадает с возрастом, определенным по фотографиям, не менее чем для 70% от имеющихся в профиле за последние два года. Сообщения пользователей, публикующих преимущественно старые фотографии (старше трех лет), в набор данных не включаются.

Такой метод позволяет отобрать только достоверные данные для обучения.

Итоговые датасеты описаны в п. 6.7.3.

5.2.2.3 Формирование признакового пространства

Было выделено три группы информативных признаков для текстов: базовые статистики, метрики удобочитаемости, метрики лексического разнообразия.

Для данного эксперимента тексты были разделены на две категории: авторы 18 и менее лет и авторы, которым 21 и более лет. В таблице 5.9 представлены базовые статистики для этих возрастных категорий.

Таблица 5.9 – Базовые статистики для определения возраста автора текста

Признак	Детская аудитория	Взрослая аудитория
Слова	10,94	13,66
Уникальные слова	9,11	12,60
Длинные слова	2,77	5,11
Сложные слова	0,72	1,67
Простые слова	7,20	10,41
Односложные слова	2,93	3,99
Многосложные слова	4,99	8,09
Символы	59,21	82,13
Буквы	36,89	63,15
Пробелы	7,82	12,12
Слоги	16,06	26,77
Знаки препинания	3,29	3,62

Из таблицы выше можно сделать вывод о том, что взрослая аудитория пишет более длинные и сложные предложения. Это видно по количеству символов и слов в предложении, употреблению сложных слов в тексте и активному использованию знаков препинания.

Следующая группа признаков – это индексы удобочитаемости. Данные метрики показывают, насколько сложен текст для восприятия. Обычно, данные индексы вычисляют на основе длины слов, предложений, использования уникальных слов.

Представленные ниже индексы были разработаны для английского языка, однако позже были адаптированы для русского, путем изменения коэффициентов:

1) тест Флеша – Кинкайда (FKGL) и Индекс удобочитаемости Флеша (FRE):

$$FKGL = 0,49\left(\frac{N_{WORDS}}{N_{SENTENCES}}\right) + 7,3\left(\frac{N_{SYLLABLES}}{N_{WORDS}}\right) - 16,59, \tag{5.14}$$

$$FRE = 206,835 - 1,3\left(\frac{N_{WORDS}}{N_{SENTENCES}}\right) - 60,1\left(\frac{N_{SYLLABLES}}{N_{WORDS}}\right), \tag{5.15}$$

где N_{WORDS} – количество слов в тексте; $N_{SENTENCES}$ – количество предложений в тексте; $N_{SYLLABLES}$ – количество символов в тексте.

2) индекс SMOG показывает количество лет обучения, необходимое для понимания текста:

$$SMOG = 1,1\sqrt{\frac{64,6N_{COMPLEX\_WORDS}}{N_{SENTENCES}}} + 0,05, \tag{5.16}$$

где $N_{COMPLEX\_WORDS}$ – количество сложных слов.

3) индекс удобочитаемости LIX показывает, насколько текст сложен для чтения:

$$LIX = \frac{N_{WORDS}}{N_{SENTENCES}} + \frac{N_{LONG\_WORDS}}{N_{WORDS}}, \tag{5.17}$$

где N_{LONG_WORDS} – количество слов длиной более шести букв.

В таблице 5.10 представлены индексы удобочитаемости для двух категорий пользователей социальной сети «ВКонтакте».

Таблица 5.10 – Индексы удобочитаемости

Признак	Детская аудитория	Взрослая аудитория
Тест Флеша – Кинкайда	0,3	4,3
Индекс удобочитаемости Флеша	97	71
Индекс SMOG	5	8,4
Индекс удобочитаемости LIX	37	50

Более высокие показатели индекса удобочитаемости Флэша говорят о легкости текста. Показатель от 90 до 100 баллов, говорит о том, что текст легко воспринимается даже 11-летними людьми, от 70 до 80 – 12–13-летними. Здесь важно отметить, что, например, среднее значение индекса Флеша для журнала «Time» имеет значение 62 балла, поэтому полученный показатель 71 балла для взрослой аудитории социальной сети «ВКонтакте» можно считать адекватным.

Также отличается индекс SMOG. Детской аудитории необходимо иметь почти 5 классов образования для понимания текста, а взрослой – 8, что сопоставимо с показателями, полученными с помощью индекса удобочитаемости Флеша.

Показатель индекса LIX для текстов детской аудитории можно интерпретировать как простые тексты, художественную литературу, газетные статьи, а для взрослой аудитории – как тексты средней сложности, журнальные статьи.

В целом на основе значений индексов удобочитаемости можно сделать вывод, что тексты взрослой аудитории сложнее для восприятия, чем детские.

Последняя группа признаков, коэффициенты лексического разнообразия, показывает богатство словарного запаса автора текста. К таким признакам относятся коэффициенты:

1) Type-Token Ratio (TTR):

$$TTR = \frac{N_{LEXEMES}}{N_{WORDS}}. \quad (5.18)$$

2) Root Type-Token Ratio (RTTR):

$$RTTR = \frac{N_{LEXEMES}}{\sqrt{N_{WORDS}}}. \quad (5.19)$$

3) Corrected Type-Token Ratio (CTTR):

$$CTTR = \frac{N_{LEXEMES}}{\sqrt{2N_{WORDS}}}. \quad (5.20)$$

В таблице 5.11 представлены коэффициенты лексического разнообразия для двух категорий пользователей социальной сети «ВКонтакте».

Таблица 5.11 – Коэффициенты лексического разнообразия

Признак	Детская аудитория	Взрослая аудитория
Type-Token Ratio (TTR)	0,90	0,96
Root Type-Token Ratio (RTTR)	2,75	3,21
Corrected Type-Token Ratio (CTTR)	1,94	2,27

5.2.2.4 Выбор информативных признаков текста

Для выбора наиболее информативных признаков из генеральной совокупности был применен метод ANOVA [79] (п. 2.9.1). В таблице 5.12 приведены 10 наиболее информативных признаков текста для задачи определения возраста автора.

Таблица 5.12 – Информативные признаки

Признак	*ANOVA*
Индекс удобочитаемости Флеша	264
Индекс удобочитаемости LIX	193
Тест Флеша – Кинкайда	168
Индекс SMOG	117
Длинные слова	107
Буквы	103
Слоги	96
Сложные слова	94
Многосложные слова	92
Type-Token Ratio (TTR)	91

Полученные признаки используются для обучения классификаторов SVM, NB, CNN, RNN, CRNN с целью решения задачи определения возраста автора

текста. Методы на основе fastText, BERT, XLM, RoBERTa осуществляют отбор признаков самостоятельно на скрытых слоях.

5.2.2.5 Преобразование текстов в тензорный вид

Преобразование текстов в векторный вид (в тензоры первого порядка) может осуществляться несколькими способами, показывающими наиболее выраженное позитивное влияние на модели классификации. Среди таких методов: токенизатор, осуществляющий получение эмбеддингов для соответствующей модели глубокой НС, и TF-IDF для простых НС и классических методов машинного обучения.

5.2.2.6 Обучение глубокой НС

Задача разделения текстов на возрастные группы нетривиальна. Исходя из опыта исследователей, освещенного в п. 1.2.1, наиболее эффективными архитектурами для определения возраста автора текста являются Наивный Байес, SVM, fastText, CNN, RNN, CRNN и BERT.

Для fastText были установлены скорость обучения равная 0,1, частота обновления скорости обучения – 100, размер контекстного окна – 5; число негативных образцов – 5; функция потерь – *softmax*. В архитектуру CRNN были включены: 1 рекуррентный слой LSTM и 3 сверточных слоя с фильтрами на 1, 3 и 4. В качестве ее параметров были установлены: размерность входного слоя – 128, функция оптимизации – Adam, функция потерь – бинарная кросс-энтропия и размер батча – 32.

В качестве архитектур, реализующих BERT, были выбраны многоязычный BERT (bert-base-multilingual-case), XLM (xlm-mlm-xnli15-1024) и RoBERTa (xlm-roberta-base) с их параметрами по умолчанию.

Для классических методов машинного обучения и простых архитектур НС параметры использовались по умолчанию.

5.2.2 Результаты экспериментов

5.2.2.1 Результаты эксперимента с фильтрацией данных

Данные для эксперимента с фильтрацией недостоверных данных были предварительно обработаны согласно рекомендациям методики. Данные были очищены от дубликатов, малоинформативных сообщений и спама. Предобработанные данные подлежали фильтрации, выполняемой с помощью модели компьютерного зрения VGG-Face. Такая фильтрация позволила отобрать 5,5 тысяч текстов авторов, возраст которых является достоверным.

На основе результатов фильтрации, было получено распределение авторов-пользователей «ВКонтакте» по их возрасту (рисунок 5.8). Согласно результатам выполненной фильтрации, пользователи имеют склонность искажать свой возраст в профиле. На графике видно снижение доли пользователей возрастом до 18 лет и увеличение доли пользователей старше 18 года.



Рисунок 5.8 – Распределение пользователей по возрасту до и после фильтрации

Полученное распределение неравномерно. Это связано с тем, что социальные сети не очень популярны среди людей старше 40 лет в России. В результате фильтрации количество сообщений пользователей младше 18 лет оказалось недостаточным, из-за чего было принято решение собрать дополнительные достоверные данные. В качестве таких данных были отобраны сочинения школьников-победителей Всероссийских конкурсов сочинений [306]. Возраст учеников точно известен, иные данные деперсонализированы.

5.2.2.2 Результаты экспериментов по определению возраста автора

В рамках эксперимента модели обучались на двух наборах данных. В первый были включены данные, отсеянные в результате фильтрации, во второй – данные, отобранные VGG-Face как достоверные. Во второй набор также вошли сочинения школьников, отобранные с целью расширения набора с достоверными сведениями о возрасте. Образцы в наборах были разделены на возрастные группы «до 18 лет» и «старше 21 года», а также на группы «до 18 лет», «от 21 до 27 лет» и «старше 30».

Важно отметить, что из эксперимента были исключены тексты лиц в возрасте от 18 до 21 года, ввиду того что авторский стиль в заданном возрастном диапазоне не является устойчивым и оказывает негативное влияние на разделяющую способность классификатора. Полученные результаты представлены в таблице 5.13.

Следует отметить, что реализованные подходы некорректно сравнивать с аналогами, разработанными для иностранных языков, ввиду более сложного словообразования и высокой степени омонимии в русском. В аналогичных работах для русского языка (Приложение А, таблица А.2) при схожей точности не учитывался класс текстов, написанных людьми моложе 18 лет, что имеет значение в задаче борьбы с педофилией в киберпространстве.

Таблица 5.13 – Результаты экспериментов по определению возраста

	Точность 2 возрастные группы, %		Точность 3 возрастные группы, %		Среднее время на эпоху, с.		Общее время обучения, с.	
Модель	Исх.	Фильт.	Исх.	Фильт.	Исх.	Фильт.	Исх.	Фильт.
NB	52	60	40	43	6	4	299	193
SVM	60	68	57	62	7	4	308	206
CNN	43	47	49	52	9	7	120	88
RNN	60	75	42	58	10	9	129	98
fastText	66	82	45	63	8	5	326	201
CRNN	64	82	43	61	14	9	149	98
BERT	66	80	41	60	55	47	552	581
XLM	50	50	38	46	69	63	697	636
RoBERT	50	75	41	60	48	42	481	430

Использование верифицированного набора данных, отобранных фильтром, обеспечивает повышение точности моделей в среднем на 12%. Полученная точность позволяет сделать вывод о целесообразности использования моделей компьютерного зрения для улучшения обучающей выборки за счет удаления недостоверных данных.

5.2.3 Выводы по подразделу

В рамках решения задачи определения возраста автора была разработана методика на основе модели fastText и метода фильтрации пользовательских фотографий с использованием модели VGG-face. Впервые учитывались тексты, написанные лицами младше 18 лет.

Установлено, что пользователи социальных сетей имеют тенденцию к занижению своего реального возраста. Тексты, авторами которых являются такие пользователи, оказывают негативное влияние на процесс обучения и последующую работу классификатора, поэтому необходимо проводить фильтрацию данных.

Эксперименты показали, что применение процедуры фильтрации по фотографиям моделью компьютерного зрения VGG-Face оказывает существенное влияние на распределение данных по возрасту и обеспечивает достоверность данных,

используемых для обучения и оценки модели определения возраста автора. Проведенный анализ подтверждает важность определения реального возраста пользователей социальных сетей, главным образом, для выявления педофилии.

Лучший результат был получен моделью fastText. Точность предсказания возрастной категории в случае бинарной классификации составила 82%, в случае многоклассовой – 63%. Сравнимые результаты дает CRNN с информативными признаками. Полученные результаты сопоставимы с современными подходами для определения возраста автора текста на английском и других языках.

Рекомендации по использованию методики:

1. Для формирования набора с достоверными данными о возрасте рекомендуется применение фильтрации с помощью компьютерного зрения.

2. В качестве алгоритмов компьютерного зрения для фильтрации фотографии рекомендуется использование современных предобученных моделей для определения возраста по фото, таких как VGG-Face.

3. Для качественного обучения классификатора возрастных групп рекомендуется использование сбалансированного по возрастным группам набора.

4. Для достижения высокой точности предсказания возрастной группы автора следует использовать параметры fastText: скорость обучения – 0,1, частота обновления скорости – 100, контекстное окно – 5; число негативных образцов – 5.

5.3 Методика идентификации пола и гендера автора текстовой информации, включая представителей ЛГБТ-сообщества*

Определение пола и гендера автора текстовой информации представляет собой важное направление в компьютерной лингвистике. Этот процесс включает анализ лингвистических, стилистических и контентных особенностей текста для выявления гендерных характеристик его создателя. Пол и гендер могут быть использованы в сочетании с другими атрибутами, такими как возраст, образование,

* Запрещенная в Российской Федерации экстремистская организация.

профессиональная принадлежность и культурный контекст, для формирования комплексного профиля автора.

В контексте кибербезопасности это важно для таких задач, как:

1. Превентивные меры предотвращения мошенничества и подделки личности. Понимание гендерных особенностей письма может помочь в выявлении случаев маскировки авторского стиля, когда злоумышленники пытаются имитировать стиль жертвы для проведения мошеннических операций.

2. Улучшение методов анализа социальных инженерных атак. Знание гендерных особенностей текста помогает в идентификации и анализе попыток социальной инженерии, нацеленных на получение конфиденциальной информации.

3. Расширенный анализ контекста угроз. Интеграция гендерного анализа в системы кибербезопасности позволяет более точно оценивать контекст и потенциальные мотивы кибератак, предоставляя дополнительные данные для принятия решений в области безопасности.

Важно отметить, что на начало 2025 года, для русского языка не существует методики, способной разграничивать текстовую информацию, созданную различными подгруппами ЛГБТ-сообщества*. В контексте поправок в законодательстве Российской Федерации, принятых в 2022–2024 годах (подразд. 1.2), анализ текстового контента, созданного представителями ЛГБТ-сообщества*, приобретает особое значение. Введенные поправки подчёркивают необходимость разработки автоматизированных решений для идентификации пола и гендера автора текста, которые бы учитывали актуальные юридические нормы и были применимы для решения практических задач.

5.3.1 Описание методики

На рисунке 5.9 представлена методика идентификации пола и гендера, основанная на принятии решений ансамблем SVM, CNN и BERT, обученных на признаковом пространстве, сформированном с помощью семантической кластеризации и частотных распределений триграмм символов, сглаженных методом Катца.

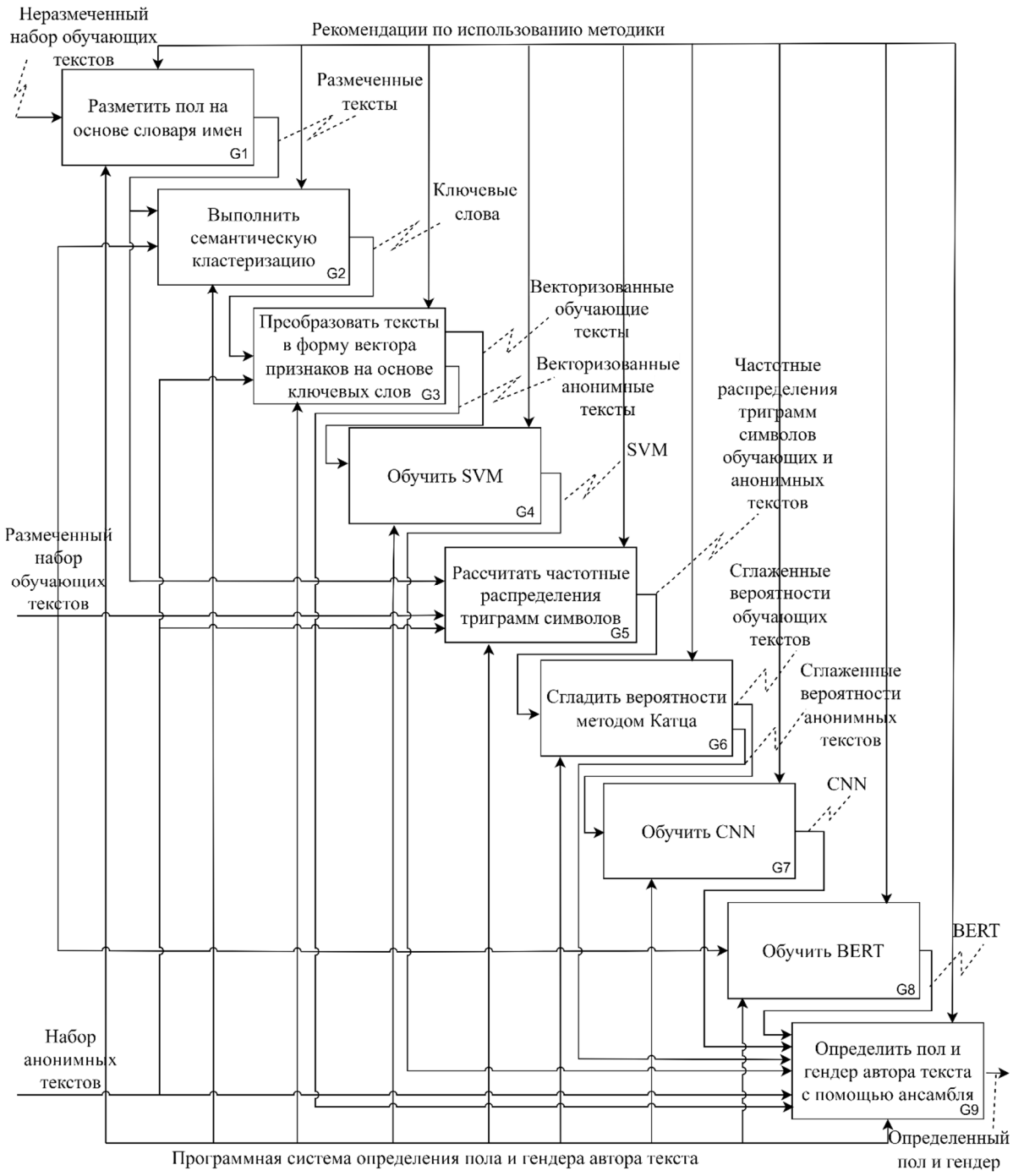


Рисунок 5.9 – Методика идентификации пола и гендера автора текста

5.3.2 Формирование признакового пространства

Для обучения моделей необходимо получить векторы мужских и женских признаков. Для их получения была проанализирована соответствующая литература по гендерной лингвистике (подразд. 1.2). Тексты, написанные мужчинами, более

проблемно-ориентированы, кратки и содержательны, в то время как женщины пишут тексты, наполненные эмоциями и выразительностью.

Для идентификации половых различий в текстах наиболее эффективными могут оказаться методы, основанные на частотности *n*-грамм. Это объясняется лингвистическими особенностями русского языка, где окончания и суффиксы прилагательных, глаголов и других частей речи могут отличаться в зависимости от гендера субъекта (например, «красивая» и «красивый», «она» и «он»). Анализ *n*-грамм позволяет выявить статистически значимые различия в использовании конкретных лексических форм мужчинами и женщинами, что может служить надёжным индикатором пола автора текста.

В то же время, при исследовании гендерной самоидентификации авторов, исключительно лингвистический подход, основанный на анализе окончаний и суффиксов, может оказаться недостаточным. Использование лексико-грамматических классов слов может в минимальной степени рефлексироваться автором и подвергаться сознательным стилистическим и иным преобразованиям – в частности для сокрытия пола автора. В этом контексте значительно большее значение приобретают семантические признаки и специфическая лексика, характерная для представителей различных гендерных групп. Например, употребление определённых фраз, которые могут быть связаны с конкретными гендерами (специфический сленг, эвфемизмы или иные лексические элементы).

В BERT тексты подаются без подготовки. Признаки для SVM сформированы на основе частотных распределений ключевых слов, полученных с помощью семантической кластеризации (п. 5.1.2), а признаки для CNN – на основе триграмм символов в наборе данных, сглаженных методом Катца (см. формулу (2.88)). Данные признаки позволяют построить квантитативную модель пола автора и тем самым решить поставленную задачу.

5.3.3 Ансамбль на основе SVM, CNN и BERT

Ансамбль, основанный на SVM, CNN и BERT, объединяет преимущества классических методов машинного обучения, глубоких и трансформенных моделей, используя разные признаковые пространства для улучшения точности.

SVM использует частотные распределения ключевых слов, полученные через семантическую кластеризацию, что позволяет модели фокусироваться на важных для смысла словах и их взаимосвязях в тексте. CNN, с другой стороны, анализирует уникальные триграммы символов, что дает возможность выявлять более тонкие и менее очевидные паттерны в структуре текста. BERT выделяет информативные признаки самостоятельно. Комбинация различных моделей и типов признаков может привести к более высокой обобщающей способности по сравнению с использованием каждой модели в отдельности. Использование ансамбля позволяет легко адаптировать модель под различные задачи (в случае 2-6 классов) без необходимости полной переработки методики. Комбинирование предсказаний от моделей часто приводит к улучшению точности классификации за счет снижения вариативности и ошибок, присущих отдельным моделям.

Параметры SVM аналогичны заданным в базовой методике определения автора текста (см. п. 3.2.4).

Для модели CNN были экспериментально определены следующие оптимальные параметры и архитектура:

1) embedding слой: размерность – 300;

2) сверточный слой: размерность слоя – 2048, размер ядра свертки – 3, функция активации – ReLU;

3) слой пакетной нормализации (BatchNormalization);

4) GlobalMaxPooling1D слой;

5) полносвязный слой: размерность – количество классов, функция активации – SoftMax;

6) количество эпох – 10;

7) размер батча – 16.

Для модели RuBERT были экспериментально определены следующие оптимальные параметры и архитектура:

1) токенизатор – rubert-basecased;

2) функция активации – ReLU;

3) коэффициент dropout – 0,2;

4) скорость обучения – 0,00001;

5) количество эпох – 10;

6) размер батча – 16;

7) количество шагов разогрева – 500;

8) константа сдвига весов – 0,01.

К результатам работы каждого из алгоритмов ансамбля применяются веса. Для BERT и CNN коэффициент составляет 0,4, для SVM – 0,2. Веса подобраны экспериментально исходя из результатов экспериментов отдельных классификаторов, представленных далее.

5.3.4 Постановка экспериментов и результаты

Было проведено 13 экспериментов, касающихся определения как половой принадлежности пользователей, так и их гендера:

1) определение пола автора среди гетеросексуальных мужчин и женщин;

2) определение принадлежности автора текста к ЛГБТ-сообществу*. В первый класс объединены все гетеросексуальные мужчины и женщины, во второй – все представители ЛГБТ*;

3) определение написан ли текст гетеросексуальным мужчиной, гетеросексуальной женщиной или представителем ЛГБТ*;

4) определение написан текст гетеросексуальным мужчиной, гетеросексуальной женщиной, гомосексуальным мужчиной или гомосексуальной женщиной;

* Запрещенная в Российской Федерации экстремистская организация.

5) определение написан ли текст гомосексуальным мужчиной или гомосексуальной женщиной;

6) определение пола автора текста среди гетеросексуальных и гомосексуальных мужчин и женщин. В первый класс объединены все мужчины, во второй – все женщины по биологическому полу;

7) определение написан ли текст гетеро- или гомосексуальным мужчиной;

8) определение написан ли текст гетеро- или гомосексуальной женщиной;

9) определение написан ли текст гетеросексуальным мужчиной, гетеросексуальной женщиной или гомосексуальным мужчиной;

10) определение написан ли текст гетеросексуальным мужчиной, гетеросексуальной женщиной или гомосексуальной женщиной;

11) определение написан ли текст гетеросексуальным мужчиной, гетеросексуальной женщиной, гомосексуальным мужчиной, гомосексуальной женщиной, бисексуалом, трансгендером;

12) конкретизация пола и гендера автора текста среди представителей ЛГБТ** (гомосексуальный мужчина, гомосексуальная женщина, бисексуал, трансгендер);

13) определение написан ли текст бисексуалом или гетеросексуальным мужчиной или гетеросексуальной женщиной.

В качестве наборов данных использовались датасеты, описанные в п. 6.7.5: посты и комментарии из мессенджера Telegram, и сообщения из социальной сети X* (бывший Twitter*). Все наборы включают как представителей движения ЛГБТ**, так и пользователей, не имеющих к нему отношения.

Результаты экспериментов, проведенных на наборе постов Telegram, приведены в таблице 5.14, на наборе комментариев Telegram – в таблице 5.15, на наборе данных X* – в таблице 5.16. Результаты получены с оптимальными параметрами классификаторов, подобранными в результате жадного поиска. Точность оценивалась при помощи 5-кратной перекрестной проверки. При проведении экспериментов все наборы данных были сбалансированы.

** Запрещенная в Российской Федерации экстремистская организация.

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

Таблица 5.14 – Результаты экспериментов по определению пола и гендера на наборе постов Telegram

Набор данных	Количество образцов в классе	Точность модели, %													
		SVM	RF	KNN	fastText	GRU	LSTM	BiLSTM	BiGRU	CNN	RuBERT	SVM+ RuBERT	CNN+ RuBERT	SVM+ CNN	SVM+CNN+ RuBERT
Муж./жен.	48 000	81±1	76±2	62±2	68±3	79±2	85±2	88±3	83±4	89±2	88±4	85±4	90±3	89±4	90±3
ЛГБТ/муж.+жен.	30 000	83±1	84±2	55±6	75±1	82±4	86±2	85±3	84±3	86±4	86±2	84±3	92±4	91±5	90±3
ЛГБТ/муж./жен.	30 000	79±7	82±3	54±1	76±2	79±3	78±3	77±2	72±3	79±3	86±2	83±2	86±3	85±5	87±3
Муж./жен./геи/лесб.	1681	71±1	66±1	60±8	64±2	73±2	75±2	80±2	77±3	81±2	85±2	83±4	87±3	83±4	85±4
Геи/лесб.	1681	89±1	83±1	85±6	86±3	88±2	84±1	88±2	85±2	90±2	88±4	90±3	92±3	88±2	92±2
Геи + муж. (гет.) /лесб+жен. (гет.)	50 172	70±4	66±2	69±4	69±2	82±3	83±2	85±2	81±4	85±2	86±2	80±3	91±2	77±3	85±3
Муж. (гет.)/геи	1681	89±1	85±2	81±3	85±3	85±1	86±3	86±2	84±1	90±6	90±1	90±3	93±3	93±5	93±3
Жен. (гет.)/лесб.	2527	91±2	88±2	84±2	86±3	89±3	88±2	89±4	90±4	92±2	95±2	95±3	96±2	95±2	95±3
Муж./жен./геи	1681	73±1	68±2	65±9	68±3	72±2	71±3	73±3	69±3	75±2	78±3	76±4	79±2	76±3	79±3
Муж./жен./лесб.	2527	78±8	72±1	69±1	72±2	74±3	76±2	77±3	72±3	79±1	83±4	83±2	85±4	79±2	85±3
Муж./жен./геи/лесб. /бисекс./транс.	1681	72±8	65±5	60±3	65±3	70±2	72±3	72±4	66±4	74±4	76±2	74±2	78±4	75±2	76±3
Геи/лесб./ бисекс./транс.	1681	87±1	78±2	75±6	79±3	83±3	82±4	82±4	84±3	85±4	86±3	88±3	87±2	85±3	88±3
Муж./жен./бисекс.	1681	75±2	65±2	62±2	64±3	71±3	70±3	71±2	74±2	75±2	88±2	84±3	90±2	74±3	88±3

Таблица 5.15 – Результаты экспериментов по определению пола и гендера на наборе комментариев Telegram

Набор данных	Количество образцов в классе	Точность модели, %													
		SVM	RF	KNN	fastText	GRU	LSTM	BiLSTM	BiGRU	CNN	RuBERT	SVM+ RuBERT	CNN+ RuBERT	SVM+ CNN	SVM+CNN+ RuBERT
Муж./жен.	60 354	67±3	66±3	51±3	64±4	75±3	83±4	83±2	69±4	85±3	92±3	78±3	90±2	90±3	84±4
ЛГБТ/муж.+жен.	96 486	72±3	72±3	56±3	70±2	78±2	85±1	86±3	80±3	88±3	91±4	86±2	93±3	90±2	93±4
ЛГБТ/муж./жен.	60 354	82±6	78±2	56±4	78±3	56±3	60±5	73±2	60±3	75±4	82±2	85±3	83±4	76±2	87±4
Муж./жен./геи/лесб.	20 212	69±4	48+6	25+2	60±1	79±2	88±3	86±4	72±4	89±2	90±2	78±3	91±3	86±2	90±3
Геи/лесб.	20 212	74±4	61±4	63±1	66±4	72±2	71±2	76±1	66±3	75±2	84±3	80±3	87±3	82±3	86±3
Геи + муж. (гет.) /лесб+жен. (гет.)	83 609	80±2	72±3	66±3	76±3	80±3	78±2	81±3	75±2	89±3	90±2	86±2	92±4	87±2	90±3
Муж. (гет.)/геи	42 594	71±3	66±6	51±2	68±3	66±4	68±3	67±2	67±2	70±2	79±3	75±2	81±2	80±3	79±2
Жен. (гет.)/лесб.	20 212	78±2	69±4	65±2	70±2	76±2	76±3	77±3	69±3	79±3	92±2	85±2	92±4	89±4	88±3
Муж./жен./геи	42 594	55±4	50±1	35±2	53±3	48±1	52±1	50±3	42±3	53±2	59±3	60±3	63±2	57±3	63±4
Муж./жен./лесб.	20 212	58±2	53±2	47±2	54±1	65±4	66±3	74±3	58±2	71±4	75±3	69±2	78±3	76±2	75±2
Муж./жен./геи/лесб. /бисекс./транс.	2913	41±2	32±1	30±3	40±3	42±3	45±2	47±3	44±2	48±3	64±2	56±4	65±3	48±3	64±4
Геи/лесб./ бисекс./транс.	2913	51±3	39±2	40±2	45±2	46±3	50±3	53±2	45±4	53±4	68±4	64±3	69±3	66±2	67±2
Муж./жен./бисекс.	2913	57±2	54±1	49±1	55±1	60±2	62±4	67±2	52±3	69±2	69±1	67±3	70±2	70±3	69±3

Таблица 5.16 – Результаты экспериментов по определению пола и гендера на наборе социальной сети X

Набор данных	Количество образцов в классе	Точность модели, %													
		SVM	RF	KNN	fastText	GRU	LSTM	BiLSTM	BiGRU	CNN	RuBERT	SVM+ RuBERT	CNN+ RuBERT	SVM+ CNN	SVM+CNN+ RuBERT
Муж./жен.	43 570	66±1	68±3	60±5	70±2	72±3	77±4	78±3	78±3	80±2	81±3	73±3	82±4	80±3	81±3
ЛГБТ/муж.+жен.	88 040	86±3	69±2	57±4	76±2	80±4	83±3	82±3	81±4	85±3	88±2	89±3	88±2	88±3	89±4
ЛГБТ/муж./жен.	43 570	63±4	55±4	38±3	67±3	71±3	76±2	77±4	76±3	79±3	82±3	76±4	84±4	80±4	82±3
Муж./жен./геи/лесб.	32 782	62±4	27±3	38±5	55±5	65±2	67±2	75±3	73±2	78±4	79±3	79±3	80±3	79±5	79±3
Геи/лесб.	31 782	75±2	60±4	66±2	75±2	73±3	73±3	81±2	82±3	83±3	88±2	82±4	89±3	85±3	88±3
Геи + муж. (гет.) /лесб+жен. (гет.)	76 252	69±2	54±4	62±2	72±3	67±3	69±3	78±3	78±4	73±3	79±3	75±2	73±3	71±4	76±2
Муж. (гет.)/геи	31 782	82±4	71±4	73±3	82±2	83±2	75±2	78±3	77±2	72±5	88±3	84±4	89±3	80±4	88±3
Жен. (гет.)/лесб.	31 782	76±2	71±2	49±4	80±1	77±3	84±3	83±3	83±3	86±2	89±3	83±4	92±3	85±2	89±3
Муж./жен./геи	31 782	60±4	59±3	50±3	69±2	66±4	73±3	72±3	73±3	76±2	75±4	75±2	77±2	76±3	78±3
Муж./жен./лесб.	36 857	64±3	57±4	35±5	78±1	68±4	74±2	75±3	75±4	78±2	78±3	75±3	81±2	80±2	80±4
Муж./жен./геи/лесб. /бисекс./транс.	30 061	41±4	28±5	33±3	41±4	41±3	41±3	41±2	37±3	42±4	53±2	49±3	54±3	45±4	52±2
Геи/лесб./ бисекс./транс.	30 061	52±4	48±3	43±2	48±3	50±2	48±2	46±4	48±2	51±3	58±3	58±2	59±3	53±3	58±4
Муж./жен./бисекс.	30 061	66±3	58±4	44±5	73±3	69±3	76±4	74±3	75±2	78±3	76±2	72±3	80±2	78±3	78±4

Общая тенденция наборов Telegram состоит в более высокой точности, полученной на наборах постов, чем комментариев. Это можно объяснить тем, что тексты публикаций созданы авторами каналов с исследуемым полом, гендером и сексуальной ориентацией или группой авторов, но содержат соответствующие признаки класса. А комментарии могут содержать недостоверную информацию, например, сообщения противников движения ЛГБТ* или ботов с реалистичным профилем, суммаризирующих текст публикации.

При анализе отдельных наборов Telegram и X** установлено, что использование ансамблей превосходит классические методы до 53%, и до 25% и 21% в сравнении с НС и глубокими НС. Кроме того, ансамбль работает до 8% лучше, чем входящие в его состав модели по отдельности. Среди классических методов отметим SVM, демонстрирующий точность, превышающую некоторые модели НС, за счет сочетания с семантической кластеризацией. Из глубоких нейросетей лучшие результаты показывает CNN, обученный на триграммах, сглаженных методом Катца, и RuBERT на основе архитектуры трансформер.

Заключительный эксперимент по определению пола и гендера автора проводился для объединенного набора данных, включающего тексты из обоих источников. В данном случае оценивалась точность только ансамблей, показавших лучшие результаты на отдельных датасетах. Результаты приведены в таблице 5.17.

В задачах разграничения по признаку биологического пола точность методики составляет 92% при классификации гетеросексуальных мужчин и женщин, и 88% при использовании смешанного набора, сочетающего гетеро- и гомосексуальных мужчин и женщин. Это подчеркивает факт установления паттернов речевого поведения, зависящих от биологического пола, а не ориентации. В случае разграничения людей одного биологического пола по признаку ориентации установлено, что гомо- и гетеросексуальные женщины используют менее схожую манеру письма, чем мужчины разной ориентации (точность 93% в случае женщин и 85 в эксперименте с мужчинами). В эксперименте, направленном на определение

* Запрещенная в Российской Федерации экстремистская организация.
** Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

представителей ЛГБТ-сообщества* и гендерной идентичности (традиционная или нетрадиционная) на основе стиля письма, получена точность 93%. Отметим и эксперимент, направленный на предсказание гендерной группы, включая четыре класса ЛГБТ* и два класса, представленных традиционными ориентациями – точность модели составила 68%.

Таблица 5.17 – Результаты экспериментов на объединенном наборе данных

Набор данных	Количество образцов в классе	Точность ансамблей, %		
		SVM+ CNN	CNN+ RuBERT	SVM+CNN +RuBERT
Муж./жен.	153 315	91±4	92±3	92±4
ЛГБТ*/муж.+жен.	336 316	91±3	93±3	92±3
ЛГБТ*/муж./жен.	153 315	74±3	82±2	82±3
Муж./жен./геи/лесб.	59 596	74±3	84±2	85±3
Геи/лесб.	59 596	84±3	89±3	90±3
Геи + муж. (гет.)/лесб + жен. (гет.)	229 372	88±2	86±2	88±3
Муж (гет.)/геи	76 057	83±2	84±4	85±2
Жен (гет.)/лесб.	59 596	86±3	93±3	91±2
Муж./жен./геи	76 057	70±4	75±3	74±4
Муж./жен./лесб.	59 596	74±4	80±2	79±3
Муж./жен./геи/лесб./бисекс./транс.	36 166	59±3	68±4	68±3
Геи/лесб./бисекс./транс.	36 166	69±3	66±3	69±3
Муж./жен./бисекс.	36 166	76±3	75±3	77±3

5.3.5 Оценка методики в сравнении с аналогами

Для оценки практической значимости разработанной методики идентификации пола и гендера было решено провести сравнение точностей методик и аналогов для русского языка, рассмотренных в аналитическом обзоре (подразд. 1.2). Результаты проведенного сравнения приведены в приложении А (таблица А.2).

* Запрещенная в Российской Федерации экстремистская организация.

Обобщая полученные данные, можно отметить, что предложенная методика – единственная, в которой рассматривается не только идентификация пола, но и гендера (включая ЛГБТ*). Методика, предложенная в диссертационной работе, достигает точности 93%, что является лучшим результатом для нетематических текстов среди представленных исследований, где точность варьируется от 44% до 85%. Одной из ключевых особенностей методики является применение семантической кластеризации и признаков на основе *n*-грамм символов и сглаживания. В отличие от других исследований, фокусирующихся на узком наборе жанров или авторов, авторская методика демонстрирует высокую эффективность на текстах пользователей с различной гендерной самоидентификацией и сексуальной ориентацией, что подчеркивает универсальность и адаптивность подхода.

5.3.6 Выводы по подразделу

Разработана методика идентификации пола и гендера автора текста, отличительной особенностью которой является сочетание семантической кластеризации и ансамблевого подхода на основе SVM, сверточной НС и трансформера BERT.

Сформированный набор данных включает публикации и комментарии мужчин и женщин, текстовые образцы представителей различных групп ЛГБТ*. Впервые для русского языка представлено автоматизированное решение для определения пола и гендера, учитывающее возможность принадлежности автора к ЛГБТ*.

Методика апробирована на разном количестве образцов. В базовых экспериментах по определению пола точность методики достигает 92%. При решении задачи детектирования принадлежности автора к ЛГБТ* точность методики – 93%. В экспериментах по разграничению гомо- и гетеросексуальных мужчин и женщин предложенный подход демонстрирует точности 85% и 93%, соответственно. Для самого сложного эксперимента – разграничение на 6 классов, 4 из которых ЛГБТ-категории*, получен результат 68%.

* Запрещенная в Российской Федерации экстремистская организация.

Общие рекомендации по использованию методики включают:

1. Наличие размеченного по полу набора данных или набора, содержащего имя/фамилию автора на русском языке или транслитом.

2. Для формирования признакового пространства SVM рекомендуется применять семантическую кластеризацию и сглаживание Катца, в качестве признаков CNN использовать триграммы символов.

3. Параметры SVM: алгоритм обучения – sequential optimization method, ядро – линейное, параметр регуляризации – 0,1, допустимый уровень ошибки – 0,00001, нормирование – включено.

4. CNN следует использовать в следующей конфигурации: embedding слой: размерность – 300; сверточный слой: размерность слоя – 2048, размер ядра свертки – 3, функция активации – ReLU; слой пакетной нормализации (BatchNormalization); GlobalMaxPooling1D слой; полносвязный слой: размерность – количество классов, функция активации – SoftMax; количество эпох – 10; размер батча – 16.

5. RuBERT следует использовать в следующей конфигурации: токенизатор – rubert-basecased; функция активации – ReLU; коэффициент dropout – 0,2; скорость обучения – 0,00001; количество эпох – 10; размер батча – 16; количество шагов разогрева – 500; константа сдвига весов – 0,01.

5.4 Методика проверки однородности текста и поиска заимствований

Данный раздел посвящен проблеме определения возможного плагиата в образовательной и академической среде. Разработанная методика направлена на выявление неоднородностей и заимствований, а также факта использования генеративных моделей при создании текста.

5.4.1 Описание методики

Методика проверки однородности текста и поиска заимствований основывается на методике открытой атрибуции (см. подразд. 3.3). Этапы представлены на рисунке 5.10.

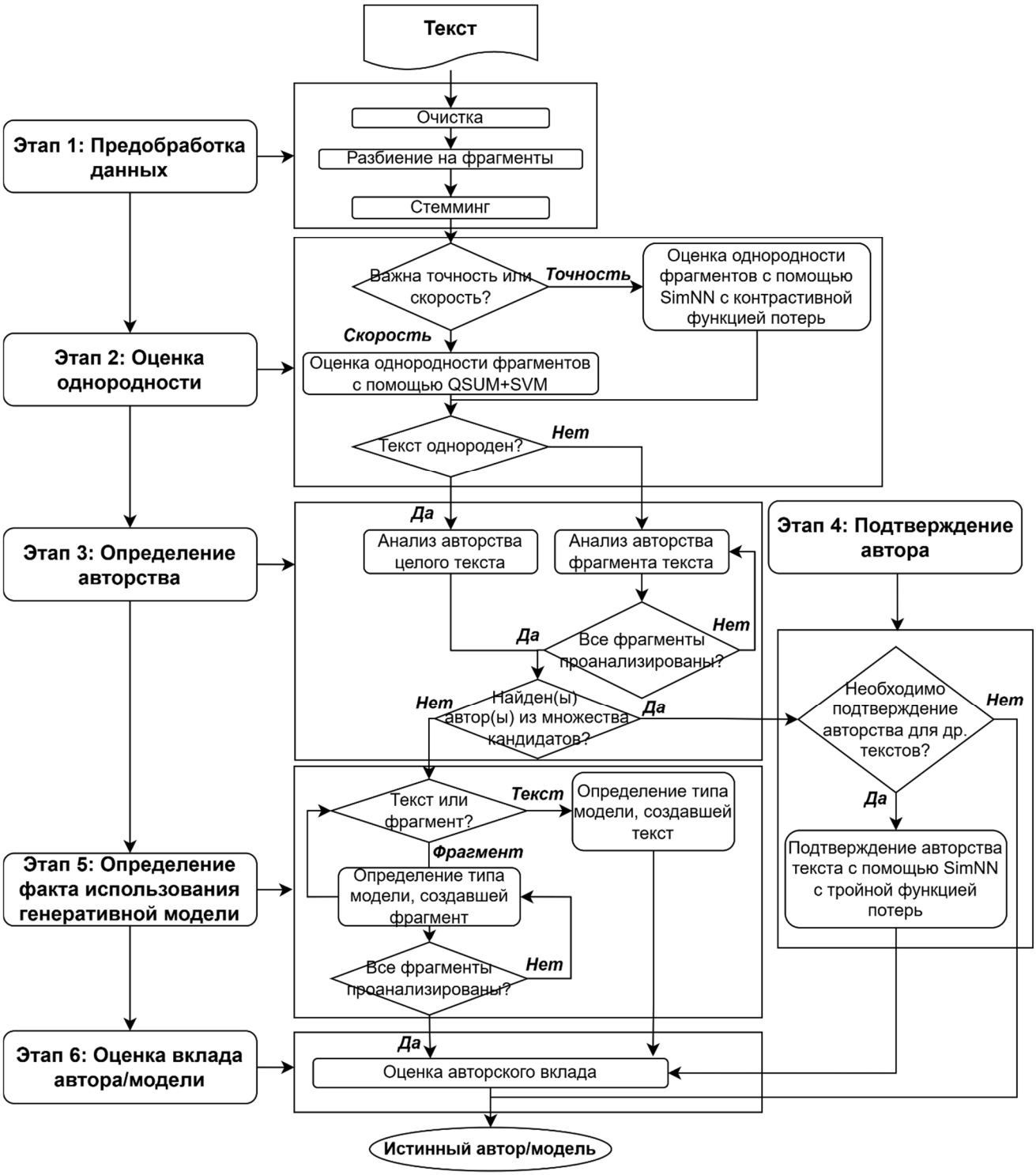


Рисунок 5.10 – Методика проверки однородности текста и поиска плагиата

На **этапе предобработки** выполняется очистка текста от персональных данных авторов, специальных символов, знаков пунктуации и стоп-слов. Затем выполняется разделение текста на фрагменты размерностью в 25 предложений [295]. Последним шагом является приведение текста к нижнему регистру и к нормальной форме.

На этапе **оценки однородности** происходит выявление неоднородных фрагментов текста. В качестве подходов предлагается использование:

1) метода QSUM совместно с SVM для случаев, где приоритетной является скорость;

2) метода на основе SimNN для случаев, где приоритетной является точность.

Для описанной задачи метод QSUM, был модифицирован следующим образом [295].

1. Извлечение признаков для каждого предложения текста. В качестве таких признаков используются ключевые слова, извлекаемые BERTopic (п. 2.8.6).

2. Расчет матрицы накопительных сумм $Q = \left(Q_{1,1},...,Q_{N,K}\right)$:

$$Q_{i,j} = \sum_{i=1,j=1}^{N,K} \left(h_{i,j} - \overline{h}_j\right), \tag{5.21}$$

где $Q_{i,j}$ – значение накопительной суммы для *i*-го фрагмента S_i для *j*-й характеристики; $h_{i,j}$ – численное значение характеристики в анализируемом фрагменте S_i; $\overline{h}_j$ – среднее значение *j*-й характеристики во всем тексте; N – количество фрагментов в тексте; K – количество характеристик.

3. Регрессионный анализ. Затем происходит расчет уравнения регрессии, определяющего зависимость между характеристиками, для каждой пары строк матрицы Q и стандартной ошибки регрессии. В качестве независимой переменной используется значение накопительной суммы.

4. Выделение потенциально неоднородных фрагментов. На данном шаге формируется матрица $G = \left(G_{1,1},...,G_{N,M}\right)$, где $G_{i,j}$ может принимать значения 0 и 1. Значение, равное 0, указывает на однородность, а 1 – на вероятную неоднородность

фрагмента текста S_i, для которого величина остаточной ошибки $\left(Q_{i,j} - Q'_{i,j}\right)$ превышает в T_1 раз стандартную ошибку регрессии:

$$Se_j = \sqrt{\frac{\sum_{i=1}^{N}\left(Q_{i,j} - Q'_{i,j}\right)^2}{N-2}}, \tag{5.22}$$

где $Q_{i,j}$ – фактическое значение накопительной суммы для *i*-го фрагмента S_i для *j*-й характеристики H_j; $Q'_{i,j}$ – рассчитанное по уравнению регрессии значение накопительной суммы.

5. Голосование. Полученные ранее векторы передаются на вход одноклассового SVM. Происходит сложение итогового числа позитивных классификаций по всем парам признаков и формируется вектор оценок. Решение о неоднородности принимается по большинству голосов.

Подход, отличающийся точностью, базируется на SimNN и состоит из двух основных шагов:

1. Основой эффективного подтверждения автора является наличие эталона с достоверным авторством и качественных векторного представления. За основу взята модель контрастивного обучения, нацеленная на минимизацию различий между текстами одного автора и максимизацию между текстами разных авторов. Такой моделью является SimNN [132, 154]. Для эффективного обучения SimNN рекомендуется использование специфической функции потерь – тройной (*t_l*) или контрастивной (*c_l*). Особенность этих функций потерь состоит в обучении модели на тройке, состоящей из якоря (текста), позитивного примера (текста того же автора) и отрицательного примера (текста другого автора). Принцип работы предложенного подхода к созданию векторных представлений представлен на рисунке 5.11.

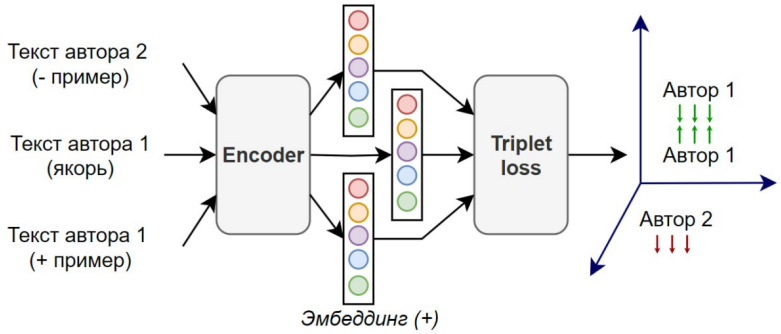


Рисунок 5.11 – Подход к созданию векторных представлений

2. На основе эмбеддингов, сформированных моделью контрастивного обучения, выполняется обучение классификационного полносвязного слоя Dense, который принимает эти эмбеддинги в качестве входных данных и выносит финальное решение о принадлежности текстов автору и однородности. При оценке однородности данным способом производится сравнение каждого фрагмента с каждым.

На этапе **определения авторства** для неоднородных фрагментов происходит идентификация автора по открытому множеству кандидатов, представленная в подразд. 3.5. Такой анализ оставляет допущение о том, что текст или фрагмент написан автором, отсутствующим в множестве кандидатов, или сгенерирован НС. В качестве метода принятия решения о принадлежности фрагмента автору используется комплекс из ГА, GRU+CNN и SVM – методов, показавших лучшие результаты при решении задачи определения авторства.

Этап **подтверждения автора** выполняется для текстов, прошедших проверку однородности. Этап направлен на подтверждение принадлежности двух и более текстов одному и тому же автору. Подтверждение автора выполняется с помощью SimNN, однако вместо предложений используются полные текст с доподлинно известным автором или моделью.

В случае, если истинный автор не был найден на этапе подтверждения, проводится анализ, направленный на **определение** факта использования **генеративных моделей** при создании текста. В качестве метода принятия решения используется гибрид GRU+CNN, показавший лучший результат согласно экспериментам, проведенным для искусственно-сгенерированных комментариев (п. 3.3.4).

Последним этапом методики является **оценка вклада автора и (или) генеративной модели** в создание анализируемого текста. Выполняется проверка авторства каждого фрагмента и суммирование частей, принадлежащих конкретному автору или модели. Итоговый вклад рассчитывается как:

$$contribution = \frac{num_a}{num_{total}} \cdot 100\%, \tag{5.23}$$

где num_a – количество фрагментов автора; num_{total} – общее количество фрагментов.

В случае, если процент заимствований составляет более 30% от общего объема, текст признается не оригинальным. Такой порог обусловлен требованиями ВАК к оригинальности диссертационных работ.

5.4.2 Результаты экспериментов по выявлению плагиата

Эксперименты, описанные далее, проводились на наборе данных из п. 6.7.6. Для определения качества оценки однородности текста были использованы научные статьи, опубликованные исследовательской группой А.С. Романова, и другие статьи с известным вкладом каждого из соавторов (453 статьи). В рамках эксперимента сравнивались точность и скорость оценки однородности фрагментов текста с помощью QSUM+SVM; и с помощью SimNN, обученной с *t_l* и *c_l* потерями, и применением ансамблями мер. Эксперименты проводились для различных по числу предложений фрагментов – от 5 до 45, так как эффективность подходов напрямую зависит от размера анализируемых фрагментов. Полученные результаты представлены в таблице 5.18.

Таблица 5.18 – Оценка точности подходов к проверке однородности

Подход	Кол-во предложений	Точность, %	Скорость, фр./с.
QSUM + SVM	5	63,2	**0,00007**
	15	81	0,00032
	25	85,5	0,00068
	35	84,1	0,00099
	45	79,5	0,00105
SimNN+*c_l*	5	75,7	0,016
	15	86,5	0,045
	25	**94**	0,081
	35	92,7	0,098
	45	90	0,103
SimNN+*t_l*	5	65,2	0,012
	15	86,1	0,042
	25	92,3	0,08
	35	91	0,091
	45	89,4	0,099

Исходя из полученных результатов, наиболее точным подходом является SimNN+*c_l*. Точность достигает 94% при размерности фрагмента в 25 предложений. Эффективность QSUM+SVM ниже на 8,5%. Однако этот подход работает на несколько порядков быстрее и может быть предпочтительным для анализа больших объемов текстовой информации.

Подход на основе SimNN, обученного с *c_l*, был использован в рамках эксперимента, посвященного определению процента однородности в текстах научных статей и диссертаций. Результаты представлены в таблице 5.19.

Таблица 5.19 – Результаты экспериментов по оценке однородности научных текстов

Домен	Число соавторов							
	–	1	2	3–5	5–7	7–10	10–15	15+
	Процент однородности текста, %							
Научные статьи	81,3	78,5	76	73,4	73,2	72,1	71	70,3
Диссертации	85,8	–	–	–	–	–	–	–

Диссертационные работы однородны на 85,8%. Такой результат обусловлен высокими требованиями к оригинальности работ и тем фактом, что диссертация пишется одним автором. Процент однородности для научных статей в случае, когда автор не имеет соавторов, составляет 81,3%. Чем больше соавторов у статьи, тем ниже однородность. Однако для 5 и более авторов разница перестает быть существенной. Это можно объяснить тем, что не все указанные в статье соавторы принимают непосредственное участие в подготовке ее текста.

Методика определения авторства для открытого множества кандидатов применялась как к текстам в их полном объеме для случая, когда текст однороден, так и к фрагментам текста с признаками заимствований. В таблице 5.20 представлены значения метрик для разного числа соавторов. Для выполнения оценки, фрагменты соавторов генерировались при помощи моделей семейства GPT на основе стилей случайных авторов из набора данных, с учетом темы статьи-оригинала. Аналогичный эксперимент был проведен для полностью искусственно-сгенерированных текстов.

Таблица 5.20 – Результаты экспериментов по определению автора научных текстов

Домен	Число соавторов							
	–	1	2	3–5	5–7	7–10	10–15	15+
	Точность, %							
Научные статьи	97,1	95	94,4	93	92	90,8	90,4	89
Диссертации	98	–	–	–	–	–	–	–
GPT-3.5	98	96,2	95	95,7	94,5	93	92,8	92,9
GPT-4	97	95,3	94,7	94	93,3	92,8	92,4	92,2

Наилучшую разделительную способность классификатор демонстрирует при отсутствии или небольшом числе соавторов. В этом случае точность достигает 98%. Наиболее простыми для идентификации оказались тексты, созданные GPT-3.5. Генеративные модели 3-го поколения имеют более выраженный структурированный стиль написания, чем модели последнего поколения.

В рамках экспериментов, посвященных подтверждению автора, оценивались различные подходы к контрастивному обучению и вычислению функции потерь. В качестве модели использовалась SimNN, в качестве функций потерь – *t_l* и *c_l*. Проверка проводилась на 2275 случайных парах текстов. В составе пары могли быть как тексты, написанные разными авторами, так и одним и тем же. Полученные результаты представлены в таблице 5.21. Лучшая точность была достигнута SimNN, обученной с *t_l*.

Таблица 5.21 – Результаты экспериментов по выбору подхода к обучению

Домен/Модель	Тройные потери	Контрастивные потери
	Точность, %	
Научные статьи	98	97,5
Диссертации	99,1	98
GPT-3.5	99,4	99,1
GPT-4	99	99

Полученные результаты указывают на особую эффективность сиамских архитектур для подтверждения авторства. Для данной комбинации была построена нормализованная матрица путаницы (рисунок 5.12).

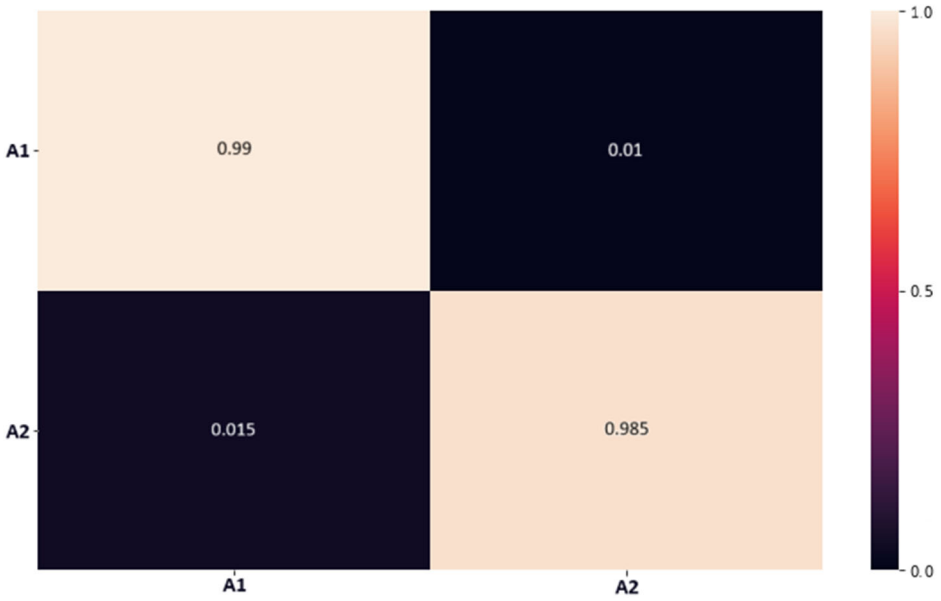


Рисунок 5.12 – Матрица путаницы SimNN, обученной с *t_l*

Классы A1 и A2 включают тексты случайных авторов из набора данных. В том случае, когда A1 пересекается в матрице конфузии с A1 или A2 пересекается с A2, считается, что оба текста принадлежат одному и тому же автору. Ложноотрицательные прогнозы в этом случае состоят в неверной классификации одного из текстов, как написанного сторонним автором. Таких прогнозов всего 2%. В случае, когда A1 пересекается с A2 и наоборот, считается, что тексты написаны двумя разными авторами. Истинно-положительные прогнозы в таком случае состоят в классификации текстов, как написанных разными авторами. Процент таких прогнозов – 98,5–99%.

Последний эксперимент был посвящен оценке вклада автора. Полученные по формуле (5.24) результаты показали, что 97% диссертационных работ имеют оригинальность более 70% и соответствуют требованиям ВАК.

В случае с научными статьями рассчитывался суммарный вклад каждого автора. Число определенных на основе неоднородных фрагментов авторов (меток авторов) не должно превышать число указанных соавторов. Полученный процент работ с совпавшим числом меток и соавторов составил 87%. В научных статьях процент меток генеративных моделей составил 9%, в диссертациях – менее 1%.

Стоит отметить, что сравнение разработанной методики с такой системой как «Антиплагиат» [231] некорректно, т.к. последняя работает по принципу полнотекстового поиска по ранее сформированным векторным представлениям для всех текстов, имеющихся в базе данных.

5.4.3 Выводы по подразделу

Разработана методика проверки однородности и поиска заимствований, отличительной особенностью которой является сочетание сиамской сети SimNN и тройной функции потерь *t_l*. Методика позволяет выявлять неоднородности, заимствования и факт использования искусственно-сгенерированных текстов, эффективно подтверждать автора двух и более текстов.

В результате эксперимента по оценке однородности выявлено, что количество соавторов существенно влияет на количество неоднородных фрагментов. Однородность диссертаций, входящих в состав набора данных, в среднем составляет 85,8%, научных статей – от 70,3% до 81,3% в зависимости от числа соавторов.

Методика была применена к полному корпусу научных текстов с целью выявления процента неоднородных, заимствованных и искусственно-сгенерированных фрагментов. Процент диссертационных работ, оригинальность которых превысила порог в 70%, составляет 97% (1% заимствований в которых – искусственная генерация), научных статей – 87% (9% – искусственная генерация).

Для достижения описанных результатов необходимо придерживаться рекомендаций по использованию методики:

1. При решении задачи по выявлению неоднородных фрагментов текста в случае, когда приоритетом является скорость предсказания, рекомендуется использование QSUM и SVM. В случае, когда скоростью можно пренебречь рекомендуется использование SimNN совместно с *c_l*.

2. В качестве характеристики для эффективной работы комбинации QSUM и SVM рекомендуется использование ключевых слов, выделенных BERTopic.

3. При решении задачи по подтверждения автора текста рекомендуется SimNN, обученная с *t_l* на протяжении не менее 50 эпох.

5.5 Выводы по главе

В результате решения важных практических задач были разработаны:

1) методика определения текстов деструктивной и экстремистской направленности;

2) методика определения возраста автора текстовой информации;

3) методика идентификации пола и гендера автора текстовой информации, включая представителей ЛГБТ-сообщества*;

4) методика проверки однородности текста и поиска заимствований.

Разработанные методики внедрены в АО «Национальный Инновационный Центр», ООО «СИБ», ООО «НТР», ООО «Сибэдж», войсковую часть 51952, ИШИТР ТПУ, ОБК УМВД России по Томской области, на экономическом факультете МГУ имени М.В. Ломоносова.

Акты внедрения представлены в приложении Г.

* Запрещенная в Российской Федерации экстремистская организация.

6 АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, НАБОРЫ ДАННЫХ ДЛЯ АНАЛИЗА ЕСТЕСТВЕННО- И ИСКУССТВЕННО-ЯЗЫКОВЫХ ТЕКСТОВ В ЗАДАЧАХ КИБЕРБЕЗОПАСНОСТИ

6.1 Программный комплекс для получения информативных признаков текста на основе семантического анализа «PyAuthorship»

6.1.1 Структура программного комплекса

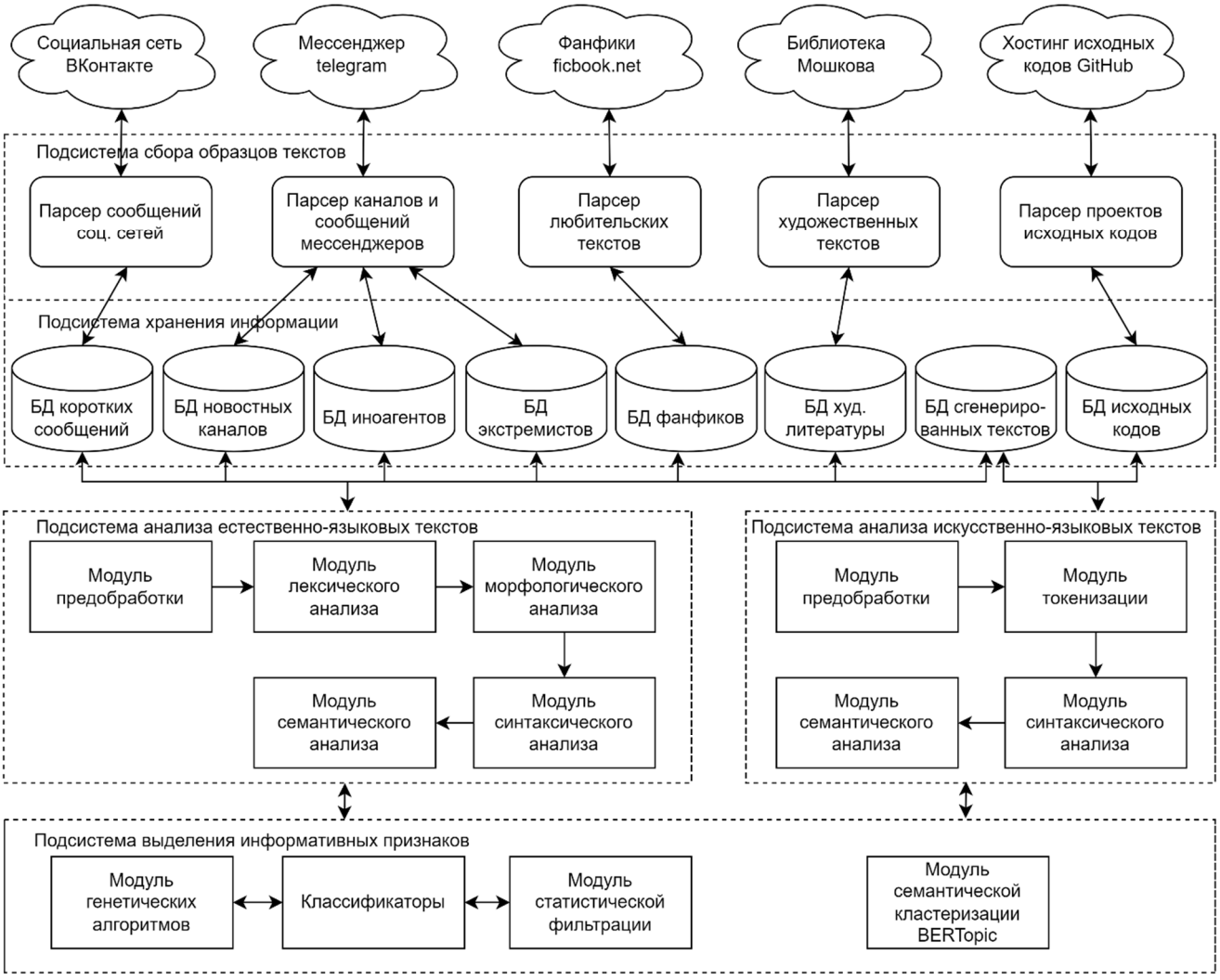


Рисунок 6.1 – Структура программного комплекса для получения информативных признаков текста

На основе методологии, разработанной в разд. 2, реализован программный комплекс для анализа естественно- и искусственно-языкового текста и решения задач кибербезопасности [300, 313, 315]. Структура программного комплекса для получения информативных признаков текста на основе семантического анализа представлена на рисунке 6.1. Также были разработаны оригинальные алгоритмы, которые легли в основу создания прикладных инструментов решения задач кибербезопасности. Описание подсистем и разработанных алгоритмов приведены в подразделах.

6.1.2 Подсистема сбора образцов текстов

В соответствии с источниками естественно- и искусственно-языковых текстов были реализованы следующие модули: парсер сообщений из социальной сети «ВКонтакте», парсер любительских текстов, парсер литературных текстов, парсер проектов исходных кодов, парсер каналов и сообщений мессенджеров.

Сбор комментариев из социальной сети «ВКонтакте» осуществлялся со страниц популярных сообществ. Сбор датасетов постов и комментариев из каналов Telegram, связанных с лицами и организациями, признанными иностранными агентами РФ и экстремистскими организациями включал следующие этапы (рисунок 6.2):

1. Идентификация каналов. Первым шагом является сбор информации о каналах Telegram, которые связаны с лицами и организациями, признанными иностранными агентами. Это может включать использование открытых источников, новостных публикаций, официальных заявлений и других доступных источников информации.

2. Создание списка каналов. На основе полученных данных необходимо составить список каналов Telegram, которые имеют прямое отношение к иностранным агентам и их деятельности.

3. Сбор новостных публикаций (постов). После создания списка каналов осуществляется сбор новостей, опубликованных в этих каналах.

4. Сбор комментариев. Происходит парсинг комментариев пользователей, обсуждающих новости, собранные на шаге 3.

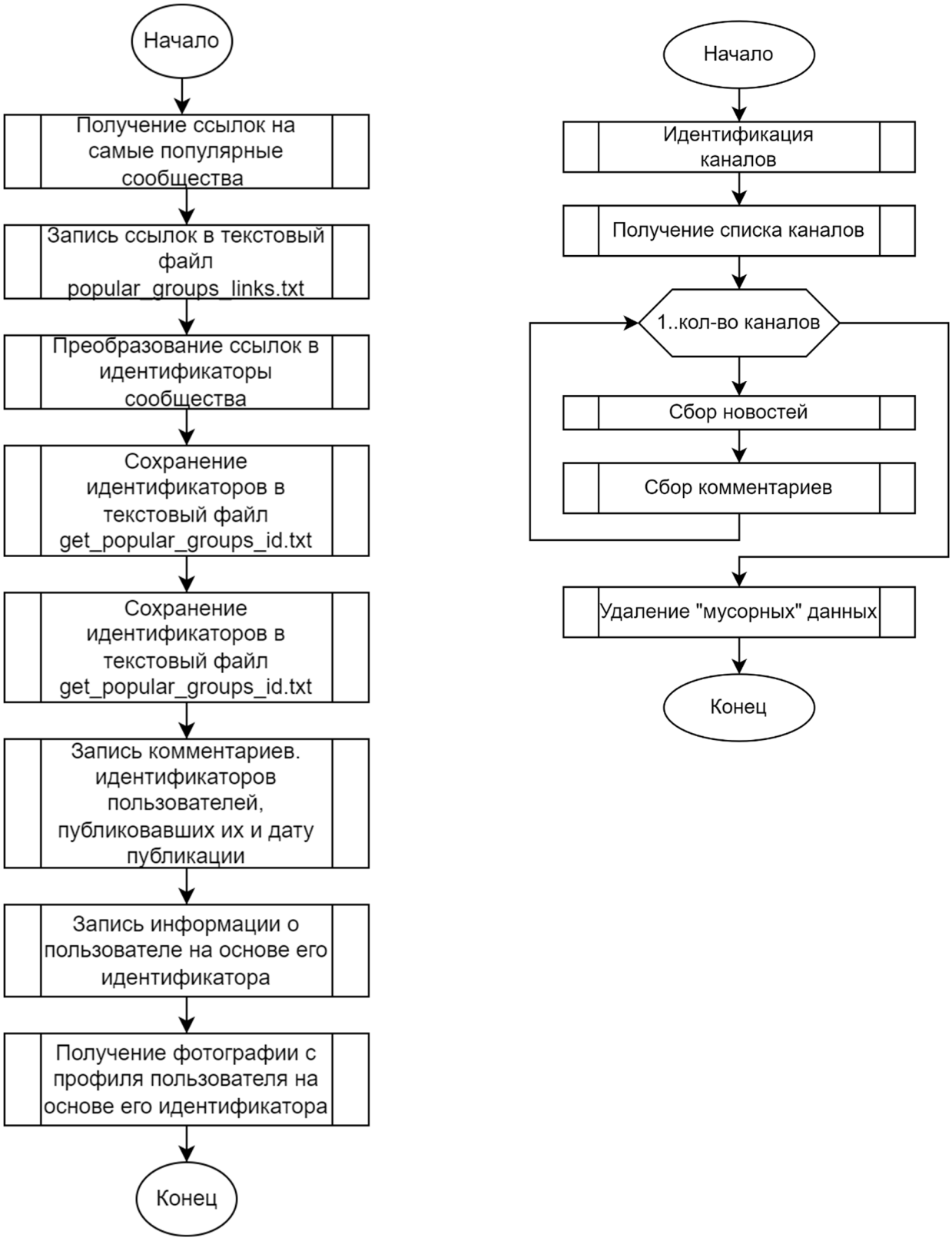


Рисунок 6.2 – Алгоритмы сбора комментариев социальной сети «ВКонтакте» (слева) и данных из каналов Telegram (справа)

6.1.3 Подсистема хранения информации

Иерархический принцип организации текста, рассмотренный в разд. 2, позволил разработать способ хранения текста в базе данных (БД) [307, 308] под управлением СУБД SQLite. Использование реляционных БД и языка SQL позволяет манипулировать обработанными текстами и их характеристиками без необходимости повторной обработки и сложного анализа. Каждая БД соответствует источнику, из которого производился сбор, и учитывает специфику типа текста: художественного, любительского, сообщений из социальных сетей и мессенджеров или исходного кода. Например, в процессе общения в социальных сетях пользователи делают публикации, которые другие пользователи могут комментировать, оценивать и делать перепубликации. Аналогично работают и каналы в мессенджере Telegram. Пользователь подписывается на канал, в котором автор канала публикует новостные посты или другой контент, подписавшиеся могут комментировать опубликованную информацию. Дополнительно можно учитывать фотографию профиля пользователя для определения пола и возраста. Описанная логика учитывается в структуре БД для хранения информации о коротких сообщениях социальных сетей и мессенджеров, представленной на рисунке 6.3.

1. Таблица Пользователи (tUsers) предназначена для хранения информации об авторах комментариев и постов в социальной сети. Важные поля, отличающие эту таблицу, содержат псевдоним (Nickname), пол (Gender), дату рождения (Birthday), популярность (Popularity), идентификатор пользователя в сети (UserID), статус, показывающий является ли автор иностранным агентом или экстремистом, или имеет отношение к организациям, имеющим эти статусы (Status).

2. Таблица Посты (tPosts) содержит информацию о публикации: дате создания публикации (Date), количестве комментариев под публикацией (Comments), количестве репостов публикации (Reposts) и количестве пользователей, оценивших публикацию (Likes) и идентификатор записи (ID_Post).

3. Таблица Фотографии (tPhotos) содержит фотографии пользователя, которые он опубликовал на своей странице в социальной сети или мессенджере.

Таблица содержит саму фотографию (Photo), дату публикации (Date), ссылку на фотографию (Link), идентификаторы записи (ID_Photo) и пользователя (ID_User).

4. Таблица Комментарии (tComments) содержит информацию о комментариях и ответах на публикацию: дате комментария (Date), количестве пользователей, которые оценили комментарий (Likes), идентификаторе комментария в социальной сети (CommentID), идентификаторе пользователя (ID_User) и идентификаторе публикации (ID_Post).

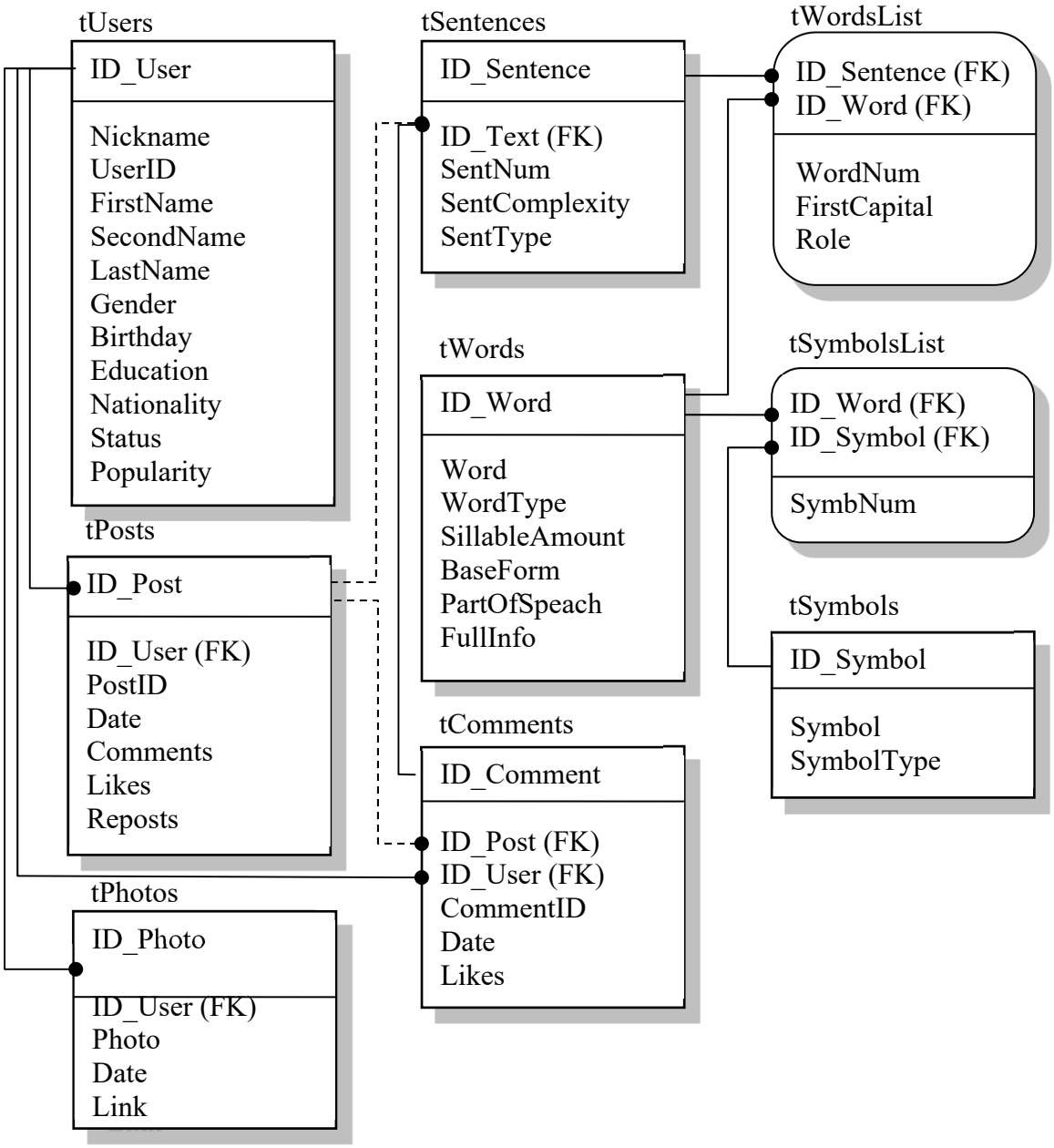


Рисунок 6.3 – Концептуальная модель БД для хранения сообщений из социальных сетей и мессенджеров

В системе контроля версий, такой как GitHub или GitLab, программист может создать один или несколько проектов и пригласить других людей принять участие в процессе его создания в качестве соавторов (контрибьюторов). Сам проект состоит из одного или нескольких файлов, содержащих исходный код. Исходный код в свою очередь представляет собой последовательность синтаксических выражений, которые в свою очередь могут содержать вложенные синтаксические выражения. Аналогом слов в языках программирования являются идентификаторы, операторы, знаки пунктуации, которые обобщенно можно назвать токенами. Токены составляют синтаксические выражения. Владелец проекта или контрибьюторы могут вносить изменения в исходный код с помощью коммитов, которые фиксируют измененные строки кода в проекте. На рисунке 6.4 представлена структура БД для хранения информации об исходных кодах.

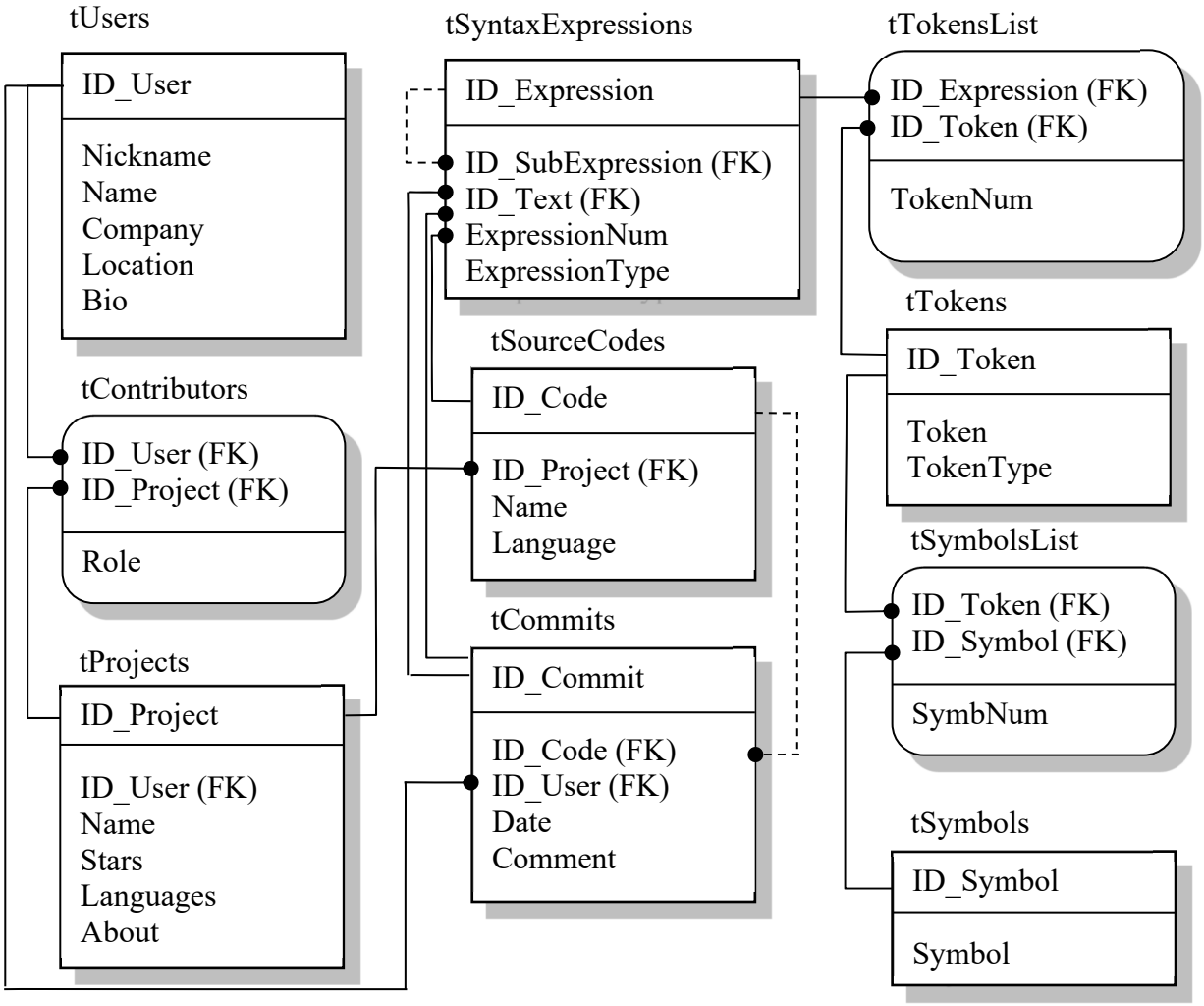


Рисунок 6.4 – Концептуальная модель БД для хранения информации об исходных кодах программ

1. Таблица Пользователи (tUsers) содержит информацию о программистах: псевдоним и имя (поля Nickname и Name), компанию, в которой он работает (Company), геолокацию, которую он указал (Location), дополнительную информацию (Bio) и идентификатор записи (ID_User).

2. Таблица Проекты (tProjects) содержит основную информацию о проекте: название (Name), перечень используемых языков программирования (Language), оценку проекта другими пользователями (Stars), краткую информацию о проекте (About) и идентификатор записи (ID_Project).

3. Таблица Контрибьюторы (tContributors) отражает информацию о том, какую роль выполняет программист в проекте (Role) и какие права в соответствии с этой ролью он имеет, а также содержит идентификаторы проекта (ID_Project) и программиста (ID_User).

4. Таблица Исходные коды (tSourceCodes) содержит название файла или модуля (Name), язык программирования, на котором он написан (Language), идентификатор проекта (ID_Project) и идентификатор записи (ID_Code).

5. Таблица Коммиты (tCommits) содержит все изменения (коммиты), внесенные в код: дату, когда выполнено изменение (Date), комментарии к нему (Comment), идентификатор пользователя, выполнившего изменение (ID_User), информацию об исходном коде (ID_Code) и идентификаторе изменения (ID_Commit).

6. Таблица Синтаксическое выражение (tSyntaxExpression) содержит информацию о синтаксическом выражении: его порядковый номер в тексте программы (ExpressionNum), тип выражения, например цикл, условие, функция (ExpressionType); идентификатор подвыражений, которые входят в данное синтаксическое выражение (ID_SubExpression), идентификатор текста, в котором используется данное выражение (ID_Code), который ссылается либо непосредственно на исходный код или коммит, и идентификатор записи (ID_Expression).

7. Таблица Токены (tTokens) предназначена для хранения операторов, идентификаторов, групп знаков препинания, значений и т.п. Содержит следующие поля: сам токен (Token), его тип (TokenType) и идентификатор записи (ID_Token).

SQL запросы для извлечения вектора интересующих характеристик из БД объединены в пакеты и хранятся в файлах. Пакеты учитывают специфику

запросов (получение результата или изменение структуры данных) и обрабатываются в системе по-разному. Алгоритмы получения файлов с данными приведены на рисунке 6.5.

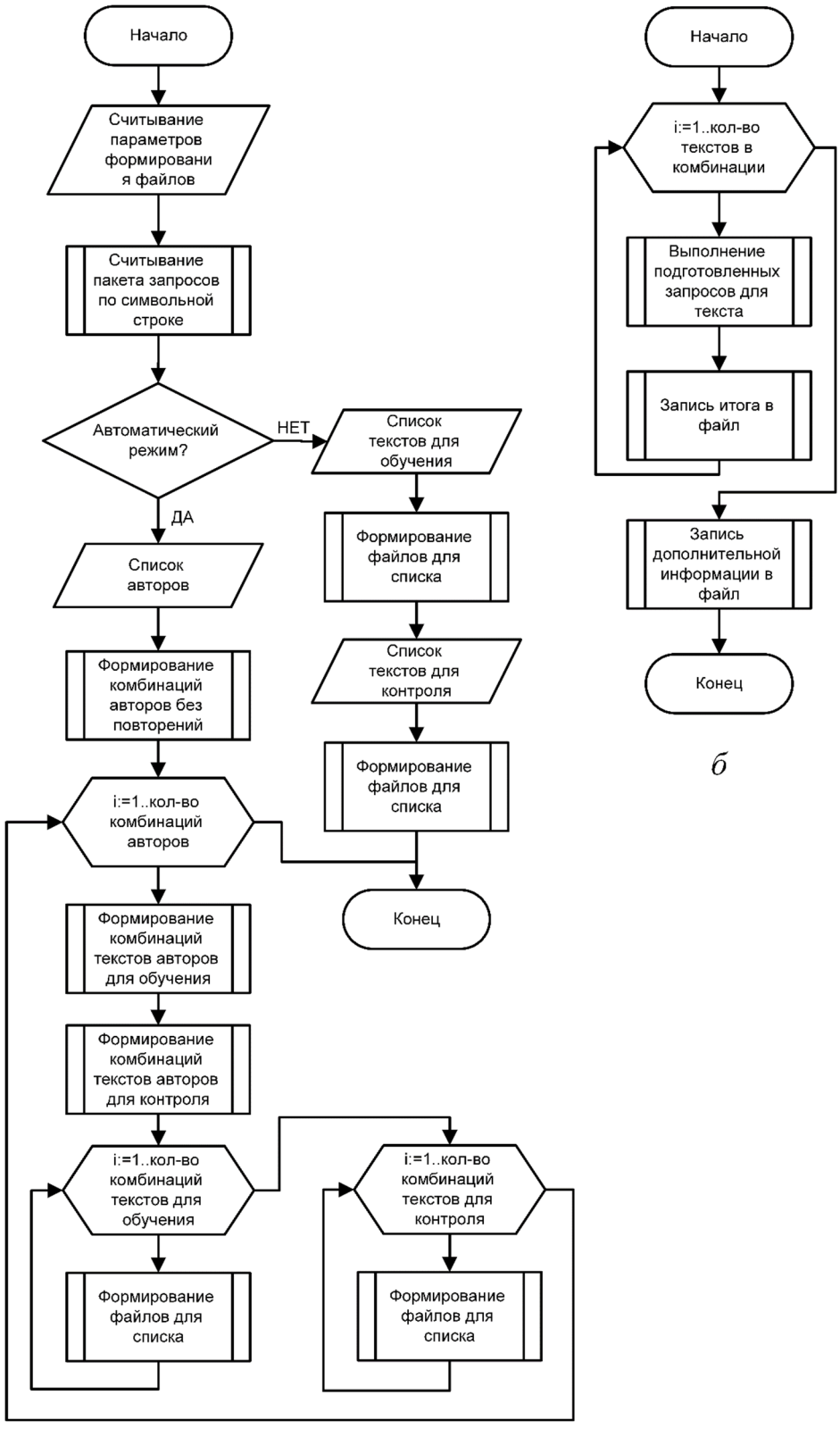


Рисунок 6.5 – Общий алгоритм получения файлов с данными (*а*) и процедура получения файлов для комбинации текстов (*б*)

6.1.4 Подсистема анализа естественного языка

В подсистеме выделяются частотные распределения, статистические характеристики длин элементов текста, синтаксические, семантические и др. признаки.

Для расчета признаков текста могут потребоваться разные способы обработки текста, например, для подсчета частот слов требуется приведение текста к нормальной форме, а для подсчета длины предложения или слова – нет. Общим для всех групп признаков является процесс базовой предобработки текста, который состоит в приведении слов документа к нижнему регистру, удалении специальных символов, очистке от букв, не являющихся элементами русского алфавита, форматирование пробельных символов. Текст, прошедший базовую предобработку, готов к извлечению различных групп признаков текста и дальнейшим преобразованиям, которые отличаются в зависимости от группы текстовых характеристик. Поэтому реализован ряд функций, позволяющих работать с каждым выбранным видом признаков и правильно их рассчитывать (таблица 6.1).

Таблица 6.1 – Описание базовых функций подсистемы анализа естественного языка

Функция	Признаки, для расчета которых используется
Векторизация текста	Семантические ключевые слова, использование категориальных признаков
Выделение ключевых слов в тексте на основе семантического анализа	Семантические признаки
Получение части речи для каждого слова	Распределение частей речи
Приведение слова к нормальной форме	Распределение слов из частотного словаря
Получение списка слов, приведенных к нормальной форме	Распределение слов из частотного словаря
Получение *n*-грамм символов внутри слова	Распределение *n*-грамм символов
Получение *n*-грамм символов в тексте	Распределение *n*-грамм символов
Получение частот знаков пунктуации	Распределение знаков пунктуации
Получение частот *n*-грамм символов	Распределение *n*-грамм символов

Функция	Признаки, для расчета которых используется
Получение частот частей речи	Распределение частей речи
Получение частот одиночных символов в тексте	Распределение одиночных символов текста
Нормализация вектора частот	Все группы признаков
Получение средней длины слова в тексте	Средняя длина слова в тексте / по автору
Получение средней длины абзаца в тексте	Средняя длина абзаца в тексте / по автору
Получение *n*-грамм слов в тексте	*n*-граммы слов
Разбиение текста на предложения	Длина предложения (средняя) в тексте / по автору
Разбиение текста на слова	Поиск самых редких / популярных слов внутри текста / по автору
Фильтрация шумовых слов	Все группы признаков
Поиск разрывных коллокаций (устойчивых сочетаний, разорванных другими словами или фразами)	Расчет различных мер схожести *n*-грамм слов и символов

Для определения границ синтаксических конструкций и блоков в языках программирования существуют четкие правила. Например, в языке С блоки кода всегда начинаются с символа «{» и заканчиваются «}», а конец инструкции завершается символом «;»; в языке программирование Python отступы в начале строки контролируют принадлежность строк кода блоку более высокого уровня, а инструкции заканчиваются символом перевода строки. Для определения границ достаточно провести простейший лексический анализ и найти соответствующие символы.

Сложнее определить конец предложения в естественном тексте. Вопросительные и восклицательные знаки, как правило, являются концом предложения всегда. Точка может означать разделитель в числе с дробной частью или служить для обозначения сокращения.

Чтобы не выполнять сложный синтаксический и семантический анализ, можно провести дополнительные проверки (рисунок 6.6):

1. Сокращения, употребляющиеся перед именами собственными. Например, г. (город), им. (имени) и другие. Предложение в этом случае продолжается.

2. Сокращения, которые могут стоять в конце предложения: т.п., др., проч., т.д., пр. Предложение закончено, если после точки идет слово с прописной буквы.

3. Сокращения, которые не могут стоять в конце предложения: т.к., т.е., т.о. Предложение в этом случае продолжается.

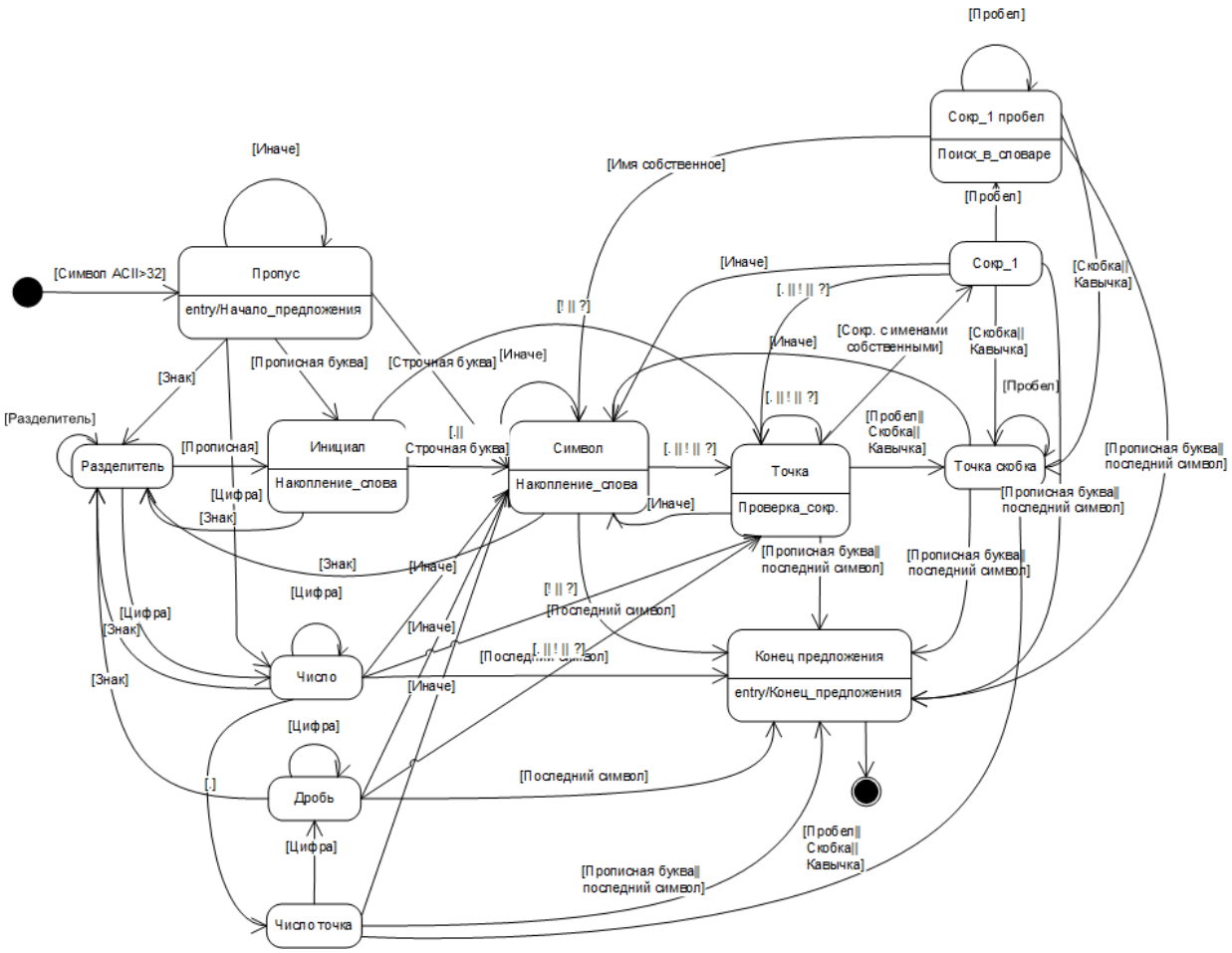


Рисунок 6.6 – Алгоритм определения границ предложений с учетом сокращений

4. Инициалы, идущие до и после фамилии. Требуется дополнительно проверить токены слева и справа от точки наличие в словаре фамилий и имен собственных. Точка, идущая после токена, состоящего из прописной буквы, не считается концом предложения. Исключением являются междометия, состоящие из одной буквы. Но обычно в этом случае они дополнительно эмоционально окрашены и употребляются с восклицательным или вопросительным знаком или многоточием.

5. Случай повторяющихся сочетаний, состоящих из точки, вопросительного или восклицательного знака. Концом предложения считается последний такой символ если следующий за ним символ – прописная буква.

6. Случай написания десятичных дробей через точку. Для этого предусмотрено дополнительное условие, согласно которому если слева и справа от точки находятся цифры, то точка не считается концом предложения.

Пошаговое применение семантического анализа представлено на рисунке 6.7.

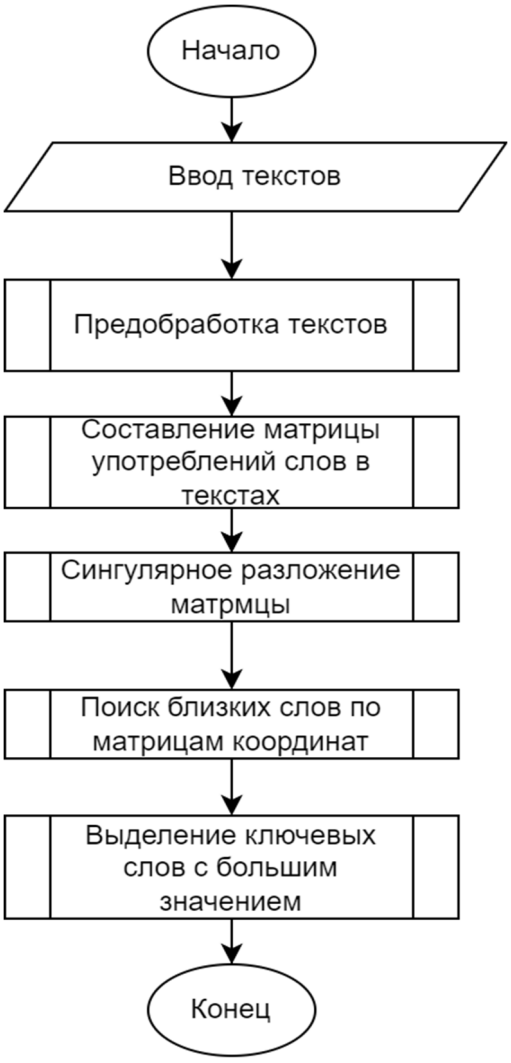


Рисунок 6.7 – Алгоритм семантического анализа текста

1. Предобработка текста. Исключение слов, встречающихся в единственном экземпляре и стоп-слов.

2. Составление матрицы нулей и единиц, соответственно представляющих отсутствие или наличие слова в документе.

3. Выполняется сингулярное разложение этой матрицы, в результате чего исходная матрица преобразует в три новые, в которых были выделены координаты документов и слов в пространстве. Сингулярное разложение выделяет ключевые составляющие матрицы, позволяя игнорировать шумы. Столбцы и строки, соответствующие меньшим сингулярным значениям, дают наименьший вклад в итоговое произведение.

4. На последнем этапе сравниваются между собой координаты документов и/или слов: те, которые находятся ближе всего составляют ключевые слова, которые необходимо найти.

Коллокации представляют собой устойчивые сочетания языка. По определению они похожи на фразеологизмы, но основная разница в том, что коллокации имеют прямой смысл, а не переносный. Использование коллокаций в качестве признаков позволяет рассчитать частоту их использования авторами, вычислить разрывы и их количественное значение по тексту и писателю. Для расчета коллокаций можно использовать следующие способы:

1. Вычисление *n*-грамм слов различной длины и расчет мер сходства между ними, вывод пар или троек слов с максимальным значением как коллокационные.

2. Поиск коллокаций с разрывами внутри *n*-грамм – в таком случае рассчитываются *n*-граммы слов, и сходство ищется между текущим словом и следующим с разрывом с несколько слов, вывод пар или троек слов с максимальным значением как коллокационные.

3. Для разрывных коллокаций используют извлечение биграмм на основе мер ассоциации и статистических критериев с различным окном.

Алгоритмы выделения коллокационных признаков представлены на рисунке 6.8.

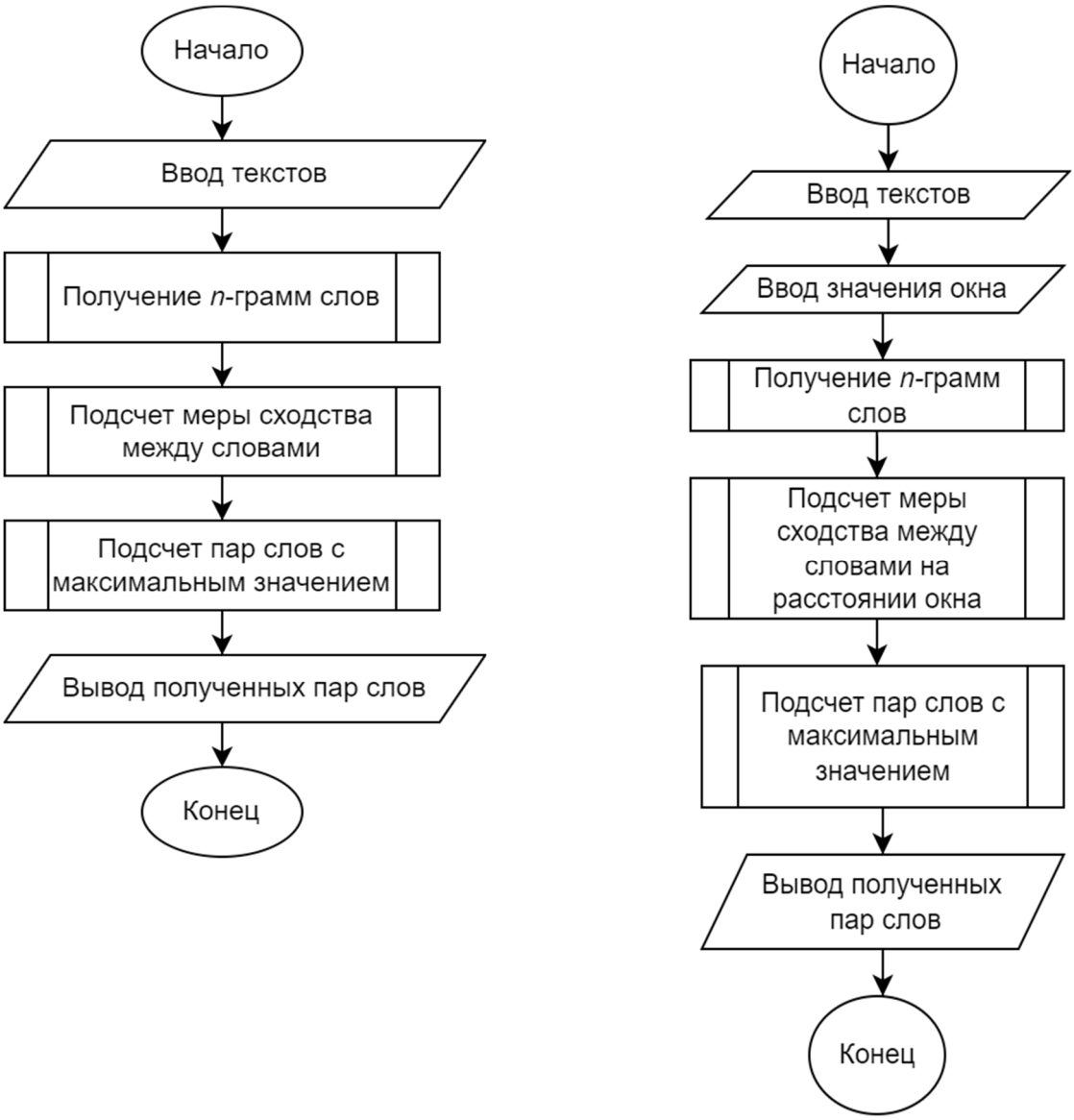


Рисунок 6.8 – Алгоритм выделения неразрывных (слева)
и разрывных (справа) коллокационных признаков

6.1.5 Подсистема анализа искусственного языка

Непосредственно анализ искусственно-языкового текста [232] во многом схож с естественно-языковым. Однако есть важные различия. Они обусловлены тем, что часть процедур, предназначенных для предобработки простого текста, могут стать причиной исчезновения важных информативных признаков для исходных кодов программ. Например, приведение текста в нижний регистр устраняет признаки, отражающие стиль создания программистом названий функций, классов и переменных, так как некоторые авторы используют в названиях не только нижний регистр, но также и верхний или даже исключительно верхний. Аналогичная ситуация и для процедуры удаления чисел, которые могут быть использованы в названиях. Форматирование пробельных символов устраняет признаки, указывающие на приверженность программиста к определенному типу пробельных символов и способу их использования. Наконец, удаление символов не английского (так как программы общепринято разрабатываются на английском языке) алфавита также не имеет смысла и, более того, оказывает негативное влияние в связи с тем, что стиль оформления комментариев, которые, как правило, пишутся программистами на родном языке, также несет в себе информативные признаки.

Извлечение признаков из искусственно-языковых текстов может осуществляться двумя путями: на основе экспертных знаний и определения множества потенциально информативных признаков обученной нейронной сетью, способной выделять их самостоятельно.

6.1.6 Подсистема выделения информативных признаков

Работа подсистемы выделения информативных признаков является финальным этапом функционирования всей системы. Получив на вход представленные в векторном виде и нормализованные данные, она производит непосредственно выделение информативных признаков. В рамках данной подсистемы функционируют

три модуля для выделения информативных признаков, а именно фильтр быстрой корреляции и глубокая НС CodeBERT.

Первый модуль подсистемы – реализует статистические методы отбора признаков. На рисунке 6.9 (слева) приведен алгоритм работы фильтра быстрой корреляции. Фильтр осуществляет работу с полным множеством доступных для анализа признаков и использует меру симметричной неопределенности для определения зависимостей между признаками.

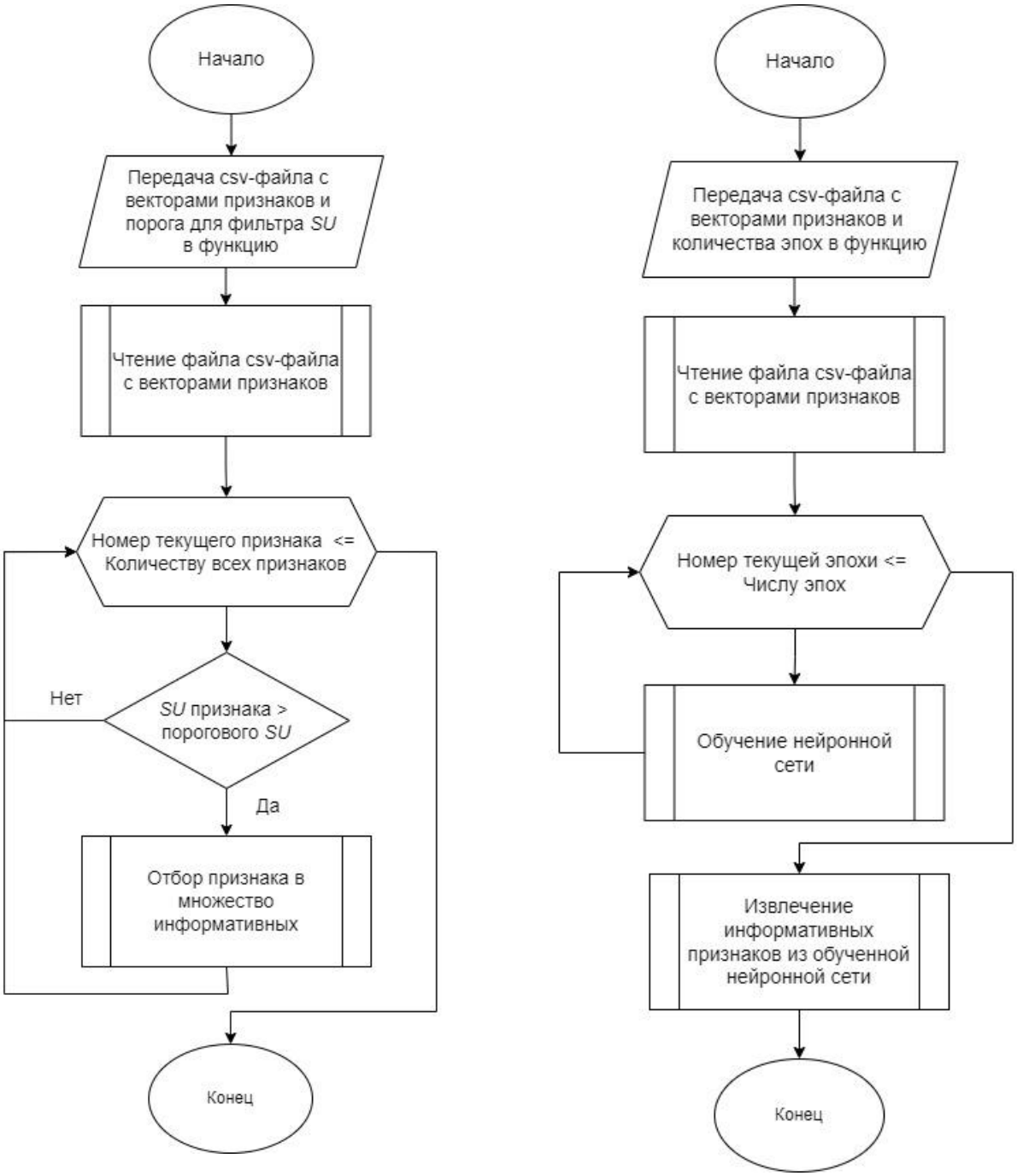


Рисунок 6.9 – Алгоритмы работы фильтра быстрой корреляции (слева) и глубокой нейронной сети (справа)

1. В модуль передаются входные параметры: csv файл с исходным множеством признаков, и пороговое значение меры симметричной неопределенности *SU*.

2. После получения признаков из файла фильтр начинает свою работу, последовательно обходя все множество имеющихся признаков.

3. В случае, если признак имеет *SU* превосходящую по величине пороговое значение, то признак отбирается в множество информативных. Цикл повторяется до тех пор, пока все признаки не будут проверены.

Второй модуль подсистемы представляет собой реализацию известных классификаторов, в частности глубоких нейронных сетей. Алгоритм работы нейронной сети представлен на рисунке 6.9 (справа).

Третий модуль подсистемы реализует генетический алгоритм отбора признаков. Хромосома представляет собой бинарный вектор признаков, а сами признаки выступают в роли генов. 1 означает, что признак информативен, а 0, что признак нужно исключить из множества информативных. Отбор признаков реализуется встраиванием генетического алгоритма в некоторый классификатор, где в качестве «пригодности» признаков используется максимальное значение точности. В качестве классификатора можно использовать любой, подходящий под решение конкретной задачи, настройка параметров классификатора также не зависит от генетического алгоритма и может быть задана произвольно. Так как операция генетического отбора затратна по времени, особенно для большого множества признаков (обычно генетический алгоритм используют для множества 12–50 элементов), то в качестве классификатора была выбрана логистическая регрессия. В качестве целевой функции можно выбрать точность и максимизировать ее или минимизировать ошибку. В данном конкретном случае была задана точность, с целью поиска ее максимального значения на сокращенном признаковом пространстве. Генетический алгоритм встраивается в цикл, в котором *n* раз происходит разделение набора данных и классификации, на каждой итерации классификация происходит на основе признаков, отобранных на прошлой итерации. Такое множество признаков составляет актуальное поколение. Критерием останова алгоритма является достижение

заданного количества итераций, а лучшим значением считается то, на котором значение целевой функции достигает максимума.

Алгоритм генетического отбора представлен на рисунке 6.10.

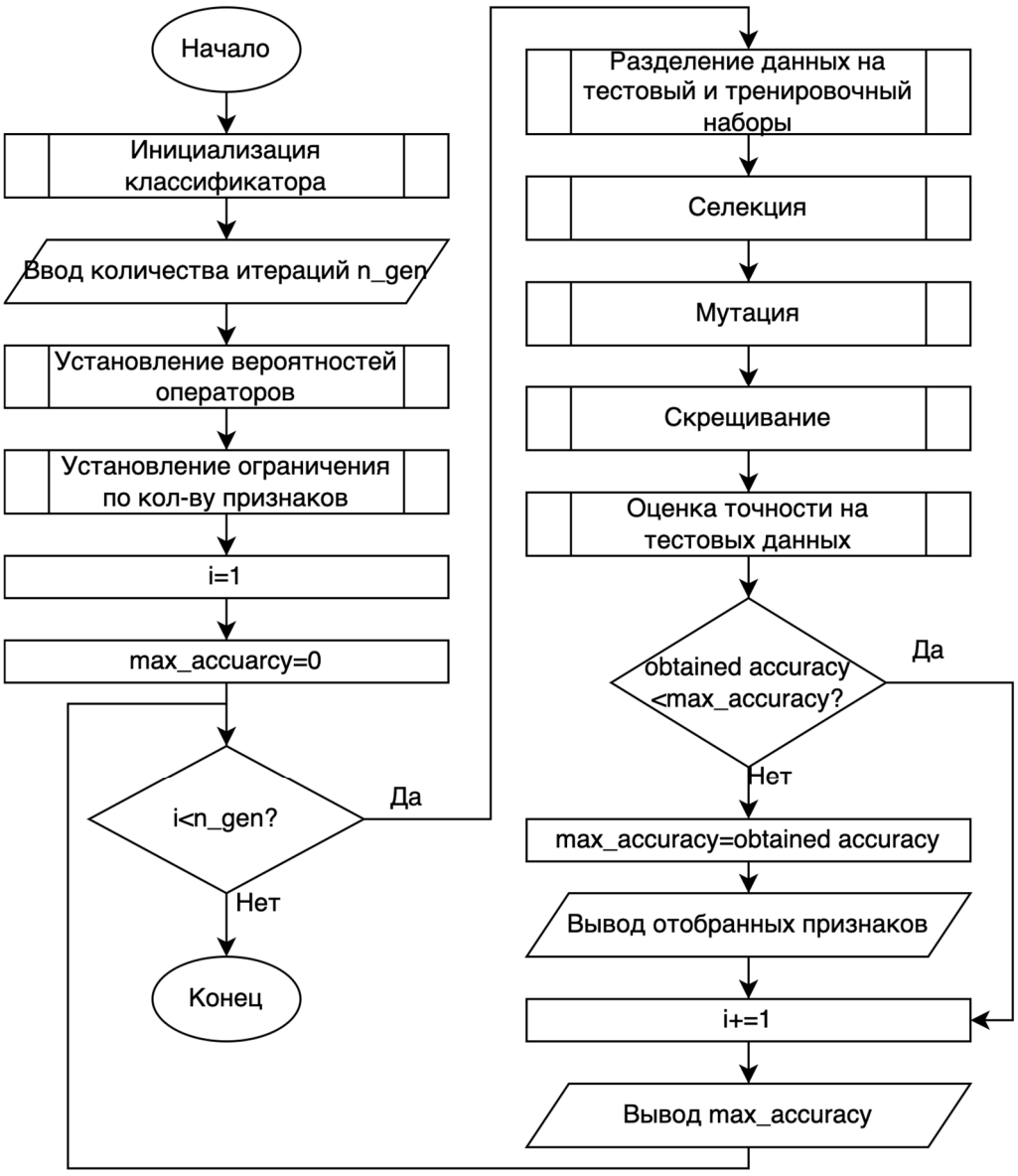


Рисунок 6.10 – Алгоритм генетического отбора

Еще один статистический метод SHAP [138] не предназначен для отбора признаков, однако это хороший инструмент для интерпретации вклада каждого индивидуального признака в прогноз модели, в результате чего SHAP позволяет оценить важность признаков. Схема работы метода представлена на рисунке 6.11.

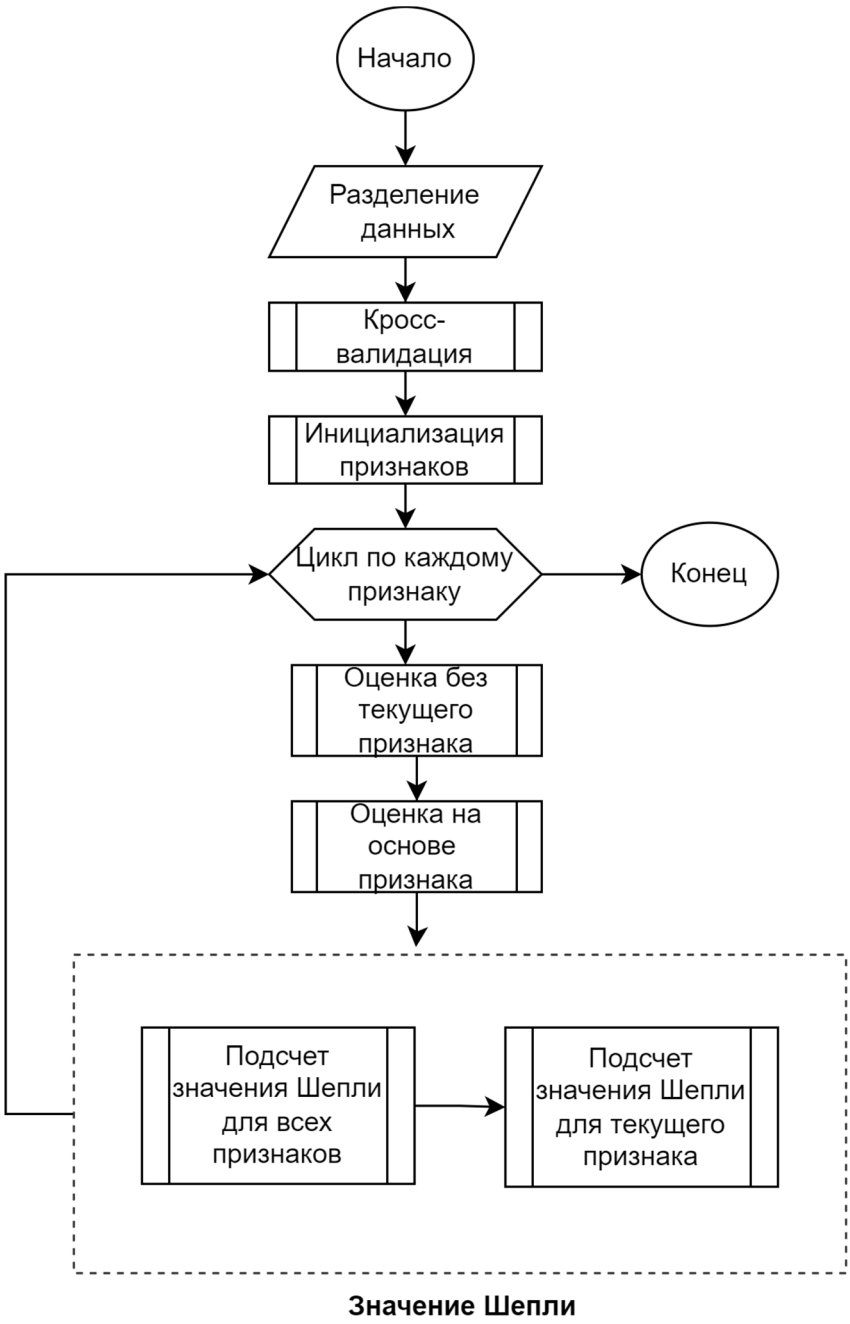


Рисунок 6.11 – Блок-схема метода SHAP

Последний модуль – модуль семантической кластеризации на основе BERTopic. Система семантической сегментации текстов состоит из 5 функциональных модулей, взаимодействующих согласно заданным пользователем аргументам (рисунок 6.12).

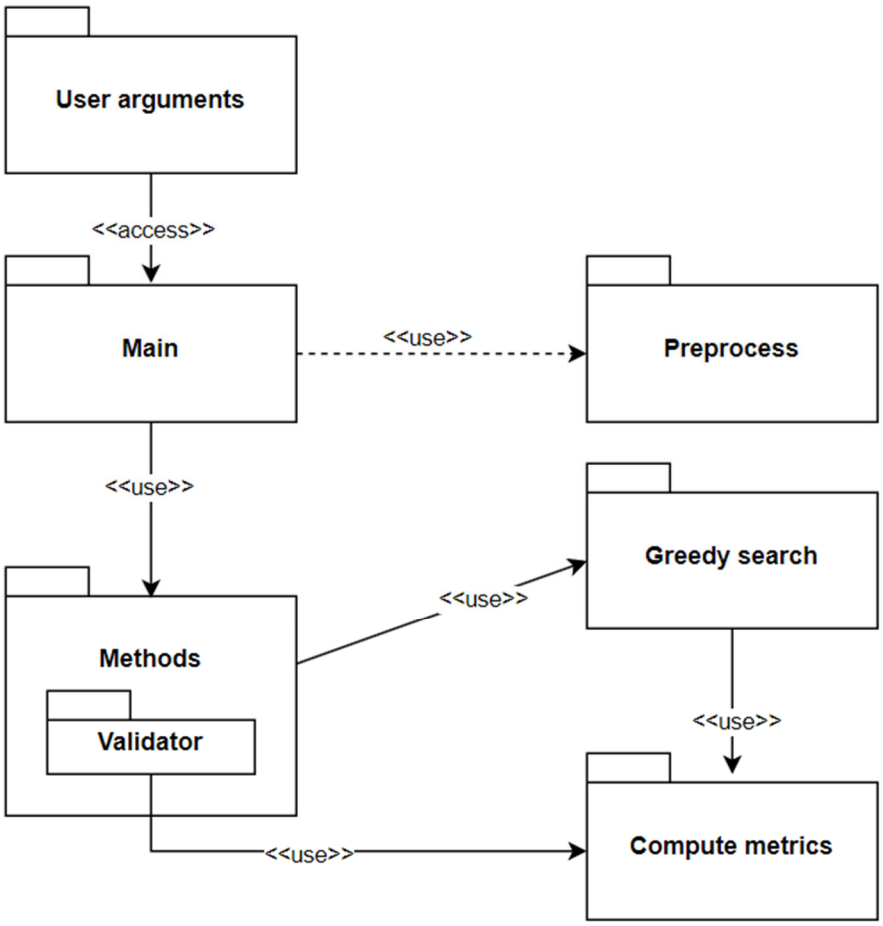


Рисунок 6.12 – UML-диаграмма пакетов семантической кластеризации

6.2 Программная система для идентификации автора письменной речи «Авторовед»

Программная система «Авторовед» [317] предназначена для исследование характеристик текста в задачах идентификации автора, определения автора спорного текста из множества потенциально возможных авторов и анализ текстовых заимствований. Структура программной системы показана на рисунке 6.13.

Подсистема сбора статистической информации состоит из модулей, осуществляющих анализ структуры текста и сбор статистической информации.

Модуль предобработки текста позволяет проводить предварительную корректировку текста.

Модуль графематического анализа осуществляет сегментацию текста и сбор статистики о выделенных единицах. Выделенные словоформы поступают в *Модуль морфологического анализа,* предназначенного для определения базовых морфологических характеристик, а также нормальной формы слова по *Морфологическому словарю*. *Модуль постморфологического анализа* позволяет избавиться от морфологической омонимии на основе частот употребления словосочетаний в русском языке.

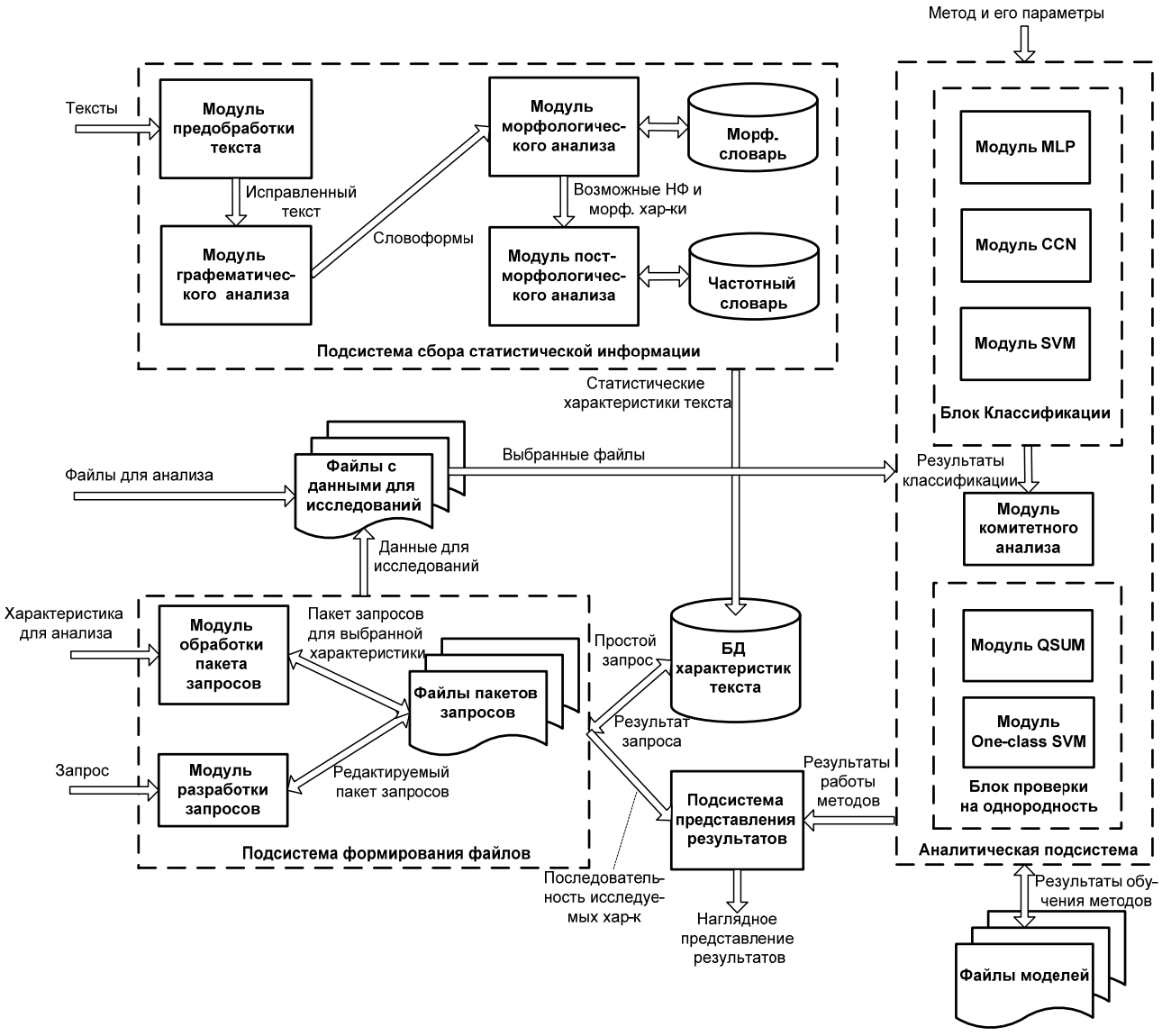


Рисунок 6.13 – Структура программной системы «Авторовед»

Собранная в результате анализа информация помещается в *Базу данных характеристик текста*. *Модуль обработки пакетов запросов* позволяет получать выборки из БД. Существует возможность расширить функционал системы за счет

добавления пользовательских пакетов в *Модуле разработки запросов*. В *Подсистеме формирования файлов для исследований* происходит подготовка извлеченных наборов данных для экспериментов.

Первый блок *Аналитической подсистемы – Блок классификации*. Он позволяет обучать классификаторы на характеристиках текста и проводить непосредственно определение автора (рисунок 6.14).

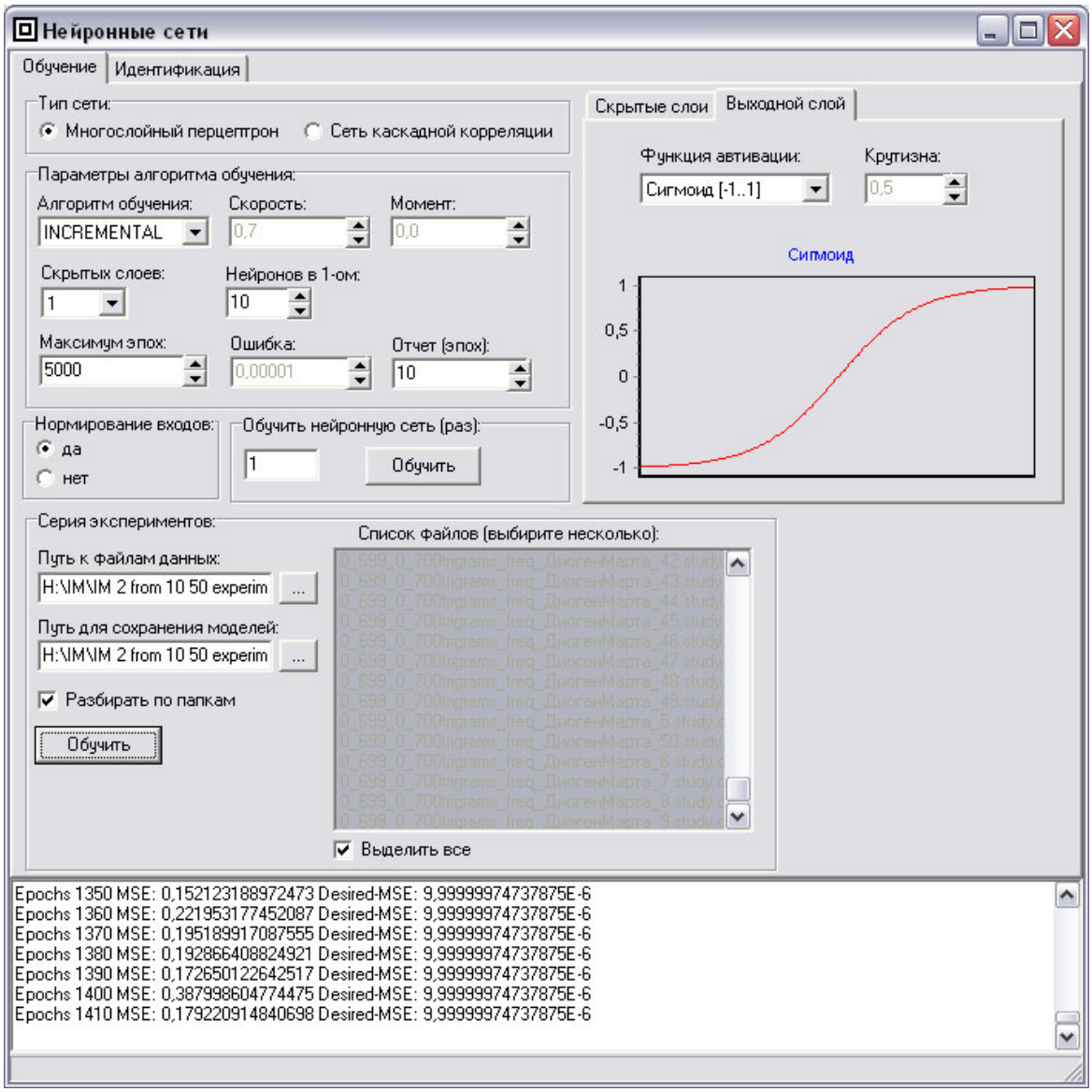


Рисунок 6.14 – Обучение нейронных сетей

В *Модуле MLP* реализуется работа с классической архитектурой многослойный перцептрон и алгоритм обучения обратного распространения ошибки. В *Модуле CCN* реализуется работа с архитектурой сеть каскадной корреляции.

Преимуществом этого метода является подбор структуры нейронной сети в процессе обучения, что приводит к сокращению временных затрат. В *Модуле SVM* реализуется работа с машиной опорных векторов. Преимуществом этого классификатора является высокая скорость обучения и простота в подборе параметров. *Модуль комитетного анализа* позволяет объединять несколько слабых классификаторов в ансамбль и повысить точность работы системы.

Блок проверки на однородность авторского стиля предназначен для анализа текстовых заимствований определения написаны ли несколько текстов одним автором. *Модуль QSUM* [292] основан на сравнении двух графиков, построенных по накопленной сумме отклонений от средних. Первый строится по отклонениям длины предложения, второй по отклонениям функциональных слов в предложении. Несовпадения двух линий могут свидетельствовать о вставке в текст предложений, отличающихся по стилю. *Модуль one-class SVM* применяется для решения задачи сравнения по образцу. Метод способен обучаться только на примерах одного класса для определения выбросов, которые затем используются как образцы второго класса для обучения классического SVM.

Подсистема представления результатов позволяет представить промежуточные данные и результаты в виде графиков, таблиц или csv файлов.

6.3 Чат-бот для определения возраста автора текстового сообщения «Age Detector»

Практическим инструментом для определения возраста автора [311] является программа «Age Detector».

Для начала работы бота необходимо ввести команду /start. После чего в мессенджере появятся две кнопки. Выбрав одну из опций, пользователь вводит текст для анализа и получает соответствующий результат в режиме переписки с ботом. «Быстрая проверка» позволит определить возраст автора введенного сообщения.

«Полная проверка» выведет значения характеристик, на основе которых принимается решение. Пример работы системы представлен на рисунке 6.15.

Программа представляет собой бота для мессенджера Telegram. Программа анализирует информативные признаки, полученные с помощью программного комплекса PyAuthroship, указывающие на возраст автора текстового сообщения [266] и реализуют методику, представленную в подразд. 5.1.

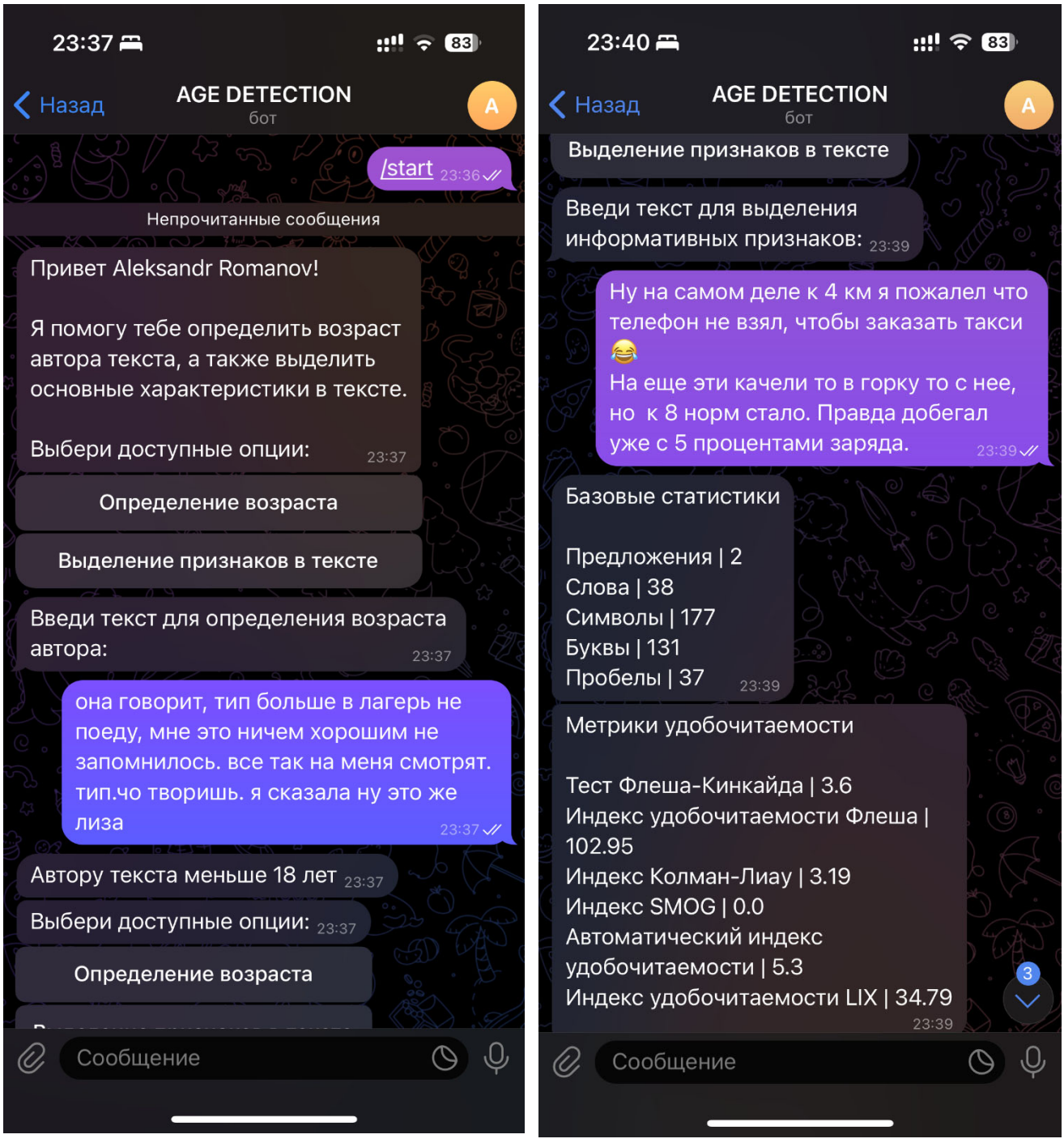


Рисунок 6.15 – Телеграмм бот для определения возраста автора текста

6.4 Система для идентификации автора исходного кода программы «CoDEtective»

Система для анализа авторства программного кода «CoDEtective» [312] представляет собой веб-приложение, упакованное в контейнер Docker, написанное на языке программирование Python. Модули системы [232], представленной на рисунке 6.16, можно сгруппировать по их функциональному назначению.

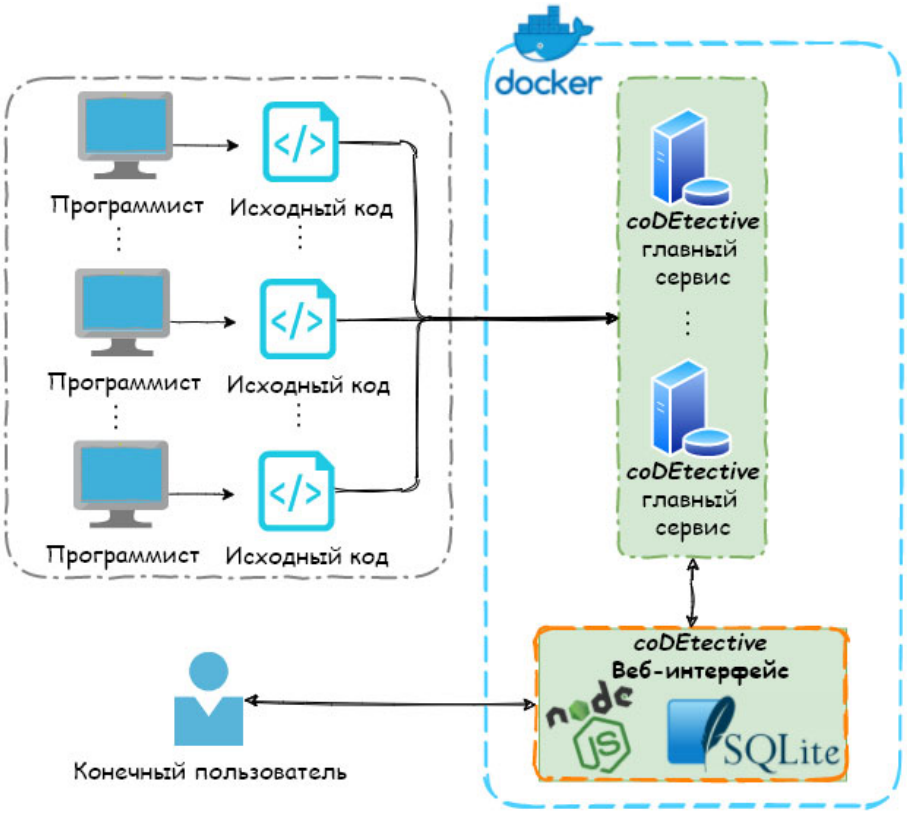


Рисунок 6.16 – Архитектура интеллектуальной системы

1) Модуль, отвечающий за преобразование исходных кодов в векторный вид и формирование на их основе корпуса для обучения. Пользователь загружает архив, содержащий каталоги с примерами кода предполагаемых программистов. На выходе из модуля формируется архив с набором данных для обучения в CSV-формате. Весь процесс описан в диаграмме на рисунке 6.17.

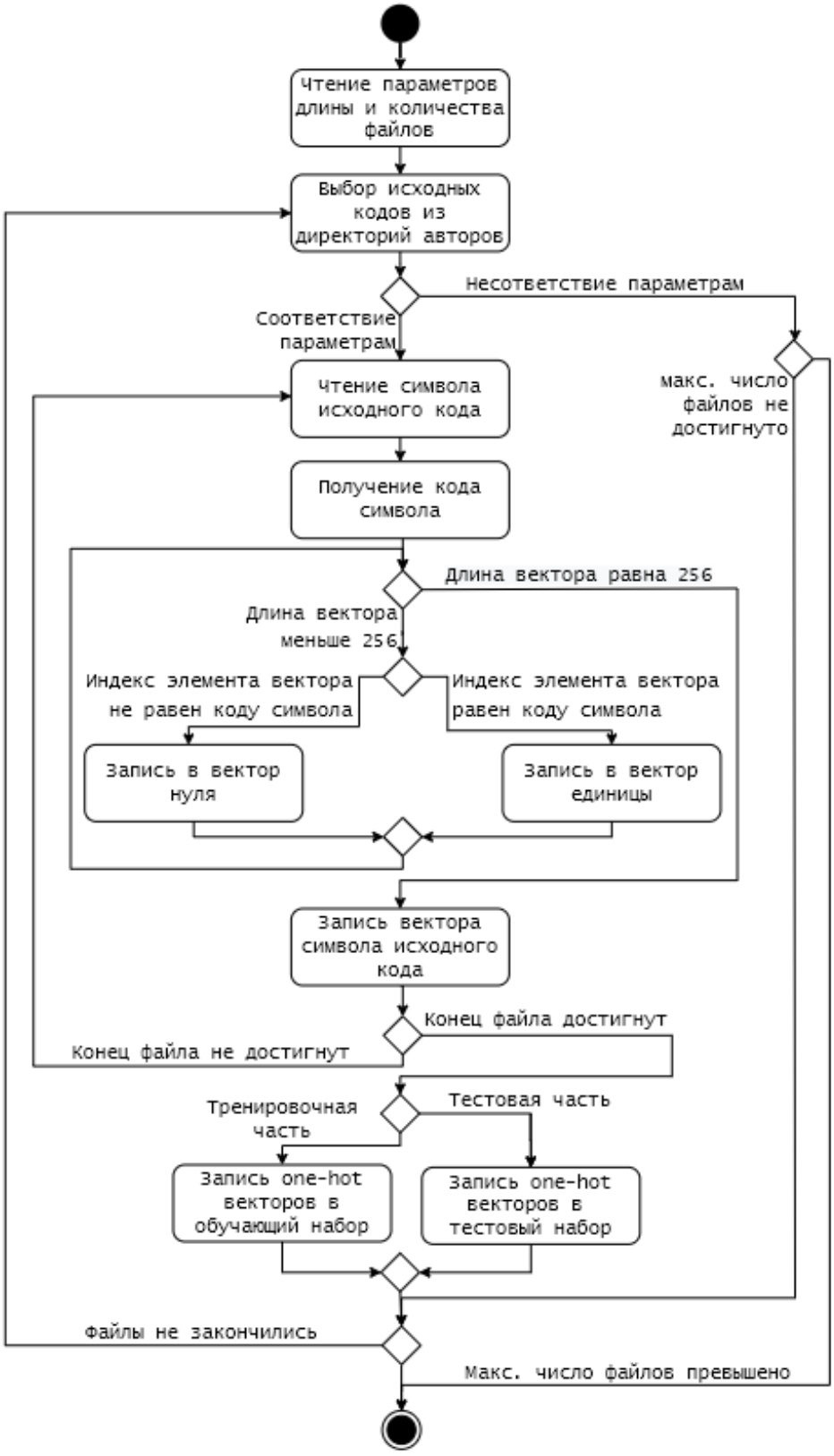


Рисунок 6.17 – UML-диаграмма деятельности формирования датасета

2) Модуль глубокого обучения, предназначенный для конфигурирования модели в соответствии с настройками и ее обучения.

3) Модуль предсказания для выполнения идентификации автора исходного кода. Используя обученную модель, пользователь может получить предсказание наиболее вероятного автора для загруженного им в систему исходного кода.

4) Модуль пользовательского интерфейса позволяет корректировать процесс работы интеллектуальной системы в зависимости от запросов и команд пользователя.

На рисунке 6.18 представлена диаграмма последовательности, демонстрирующая процесс работы с системой «CoDEtective», для сотрудников некоторой IT-компании. Диаграмма учитывает такие факторы, как использование разработчиками сторонних ресурсов (таких как StackOverflow, GitHub и пр.) и внутренних ресурсов (проектов компании), а также системы контроля версий Git. В качестве администратора и пользователя системы выступает специалист по информационной безопасности. На рисунке 6.19 представлен скриншот работы интеллектуальной системы.

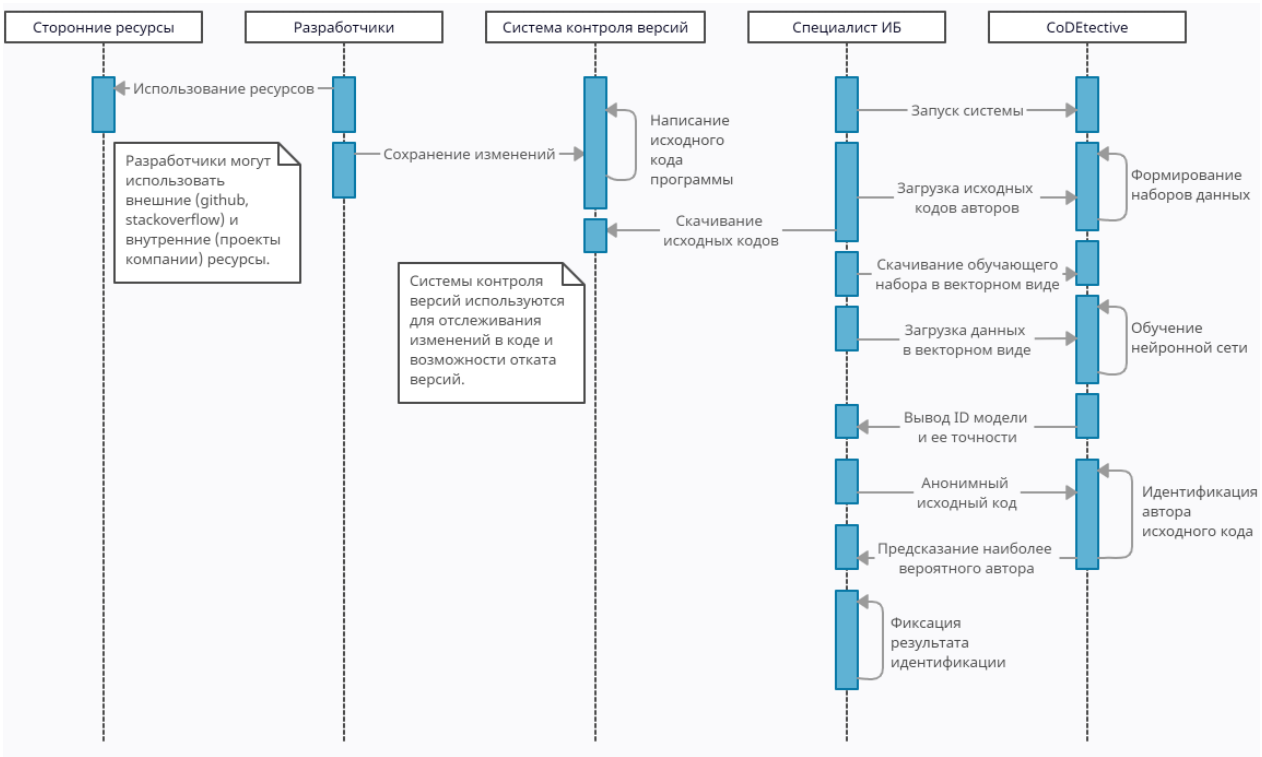


Рисунок 6.18 – UML-диаграмма последовательности работы с интеллектуальной системой «CoDEtective»

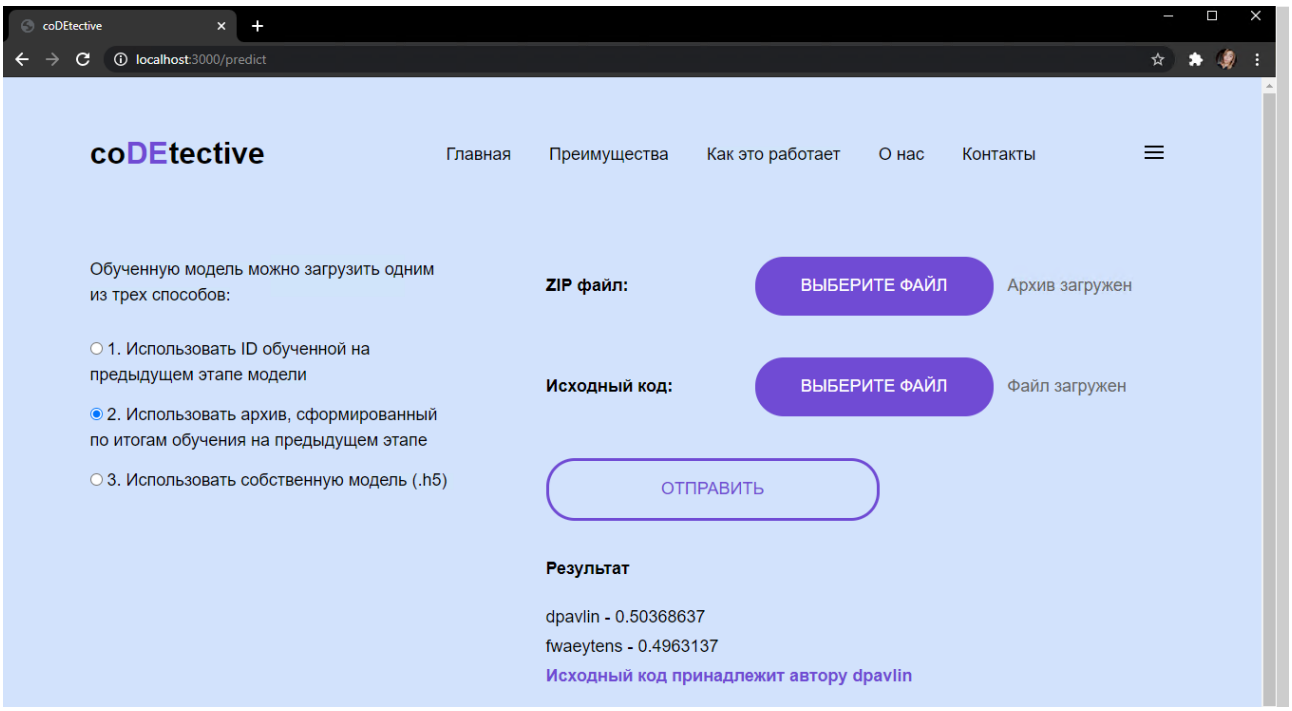


Рисунок 6.19 – Идентификация автора исходного кода

6.5 Программа для тонального анализа текста «Moodorrude»

Еще одним разработанным прикладным инструментом текстового анализа является программа для анализа настроения текста «Moodorrude» [310]. Программа предназначена для автоматизированного выявления в тексте эмоциональной оценки (положительной, отрицательной, нейтральной). Написана на языках программирования Python (серверная часть) и JavaScript (клиентская часть).

В качестве классификаторов выступают методы LinearSVC, MultinomialNB, BernoulliNB, SVM, BiLSTM. Система обучена на корпусе данных из социальной сети Twitter*, отзывах платформы Яндекс.Маркет. В программе используются словари позитивно и негативно окрашенной лексики, обсценной лексики, позитивных и негативных эмотиконов, полученные с помощью программного комплекса PyAuthorship. Система позволяет определять настроение текста с точностью 96,9% при использовании метода SVM.

Скриншот работы программы приведен на рисунке 6.20.

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

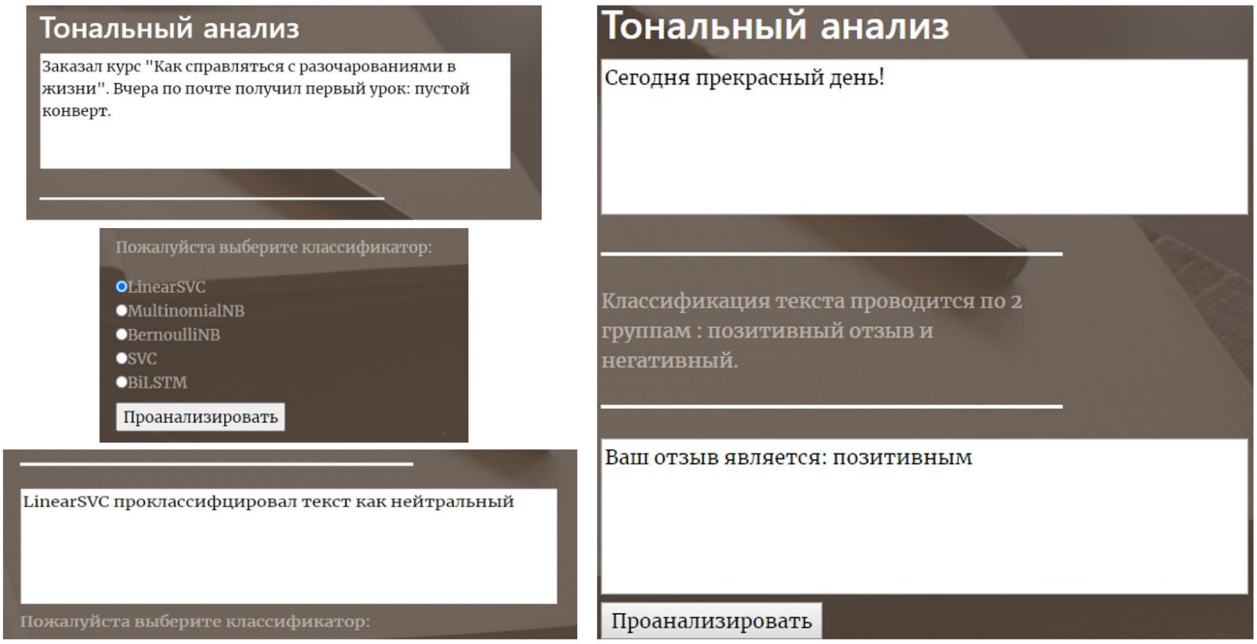


Рисунок 6.20 – Скриншот работы программы тонального анализа текста

6.6 Программный комплекс для идентификации запрещенных законодательством РФ и деструктивных текстов «PyDestructiveDetector»

Программные комплексы для идентификации текстов экстремистской направленности «PyDestructiveDetector» [314] и «ProGender» [316] созданы на базе программного комплекса PyAuthorship и предназначена для идентификации лиц, распространяющих экстремистские или созданные иностранными агентами текстовые материалы. В программном комплексе для принятия решения используются BERT, GRU+CNN и трансферное обучение.

UML диаграмма компонентов представлена на рисунке 6.21.

Предварительная обработка данных осуществляется путем вызова *dataset_formation.py.* Для вызова модуля предварительной обработки датасета, содержащего два класса, используется модуль *binary.py.* Если классов больше, используется модуль *multi_class.py.* Результатом работы модуля является сбалансированный набор данных и график, отражающий распределение данных по классам.

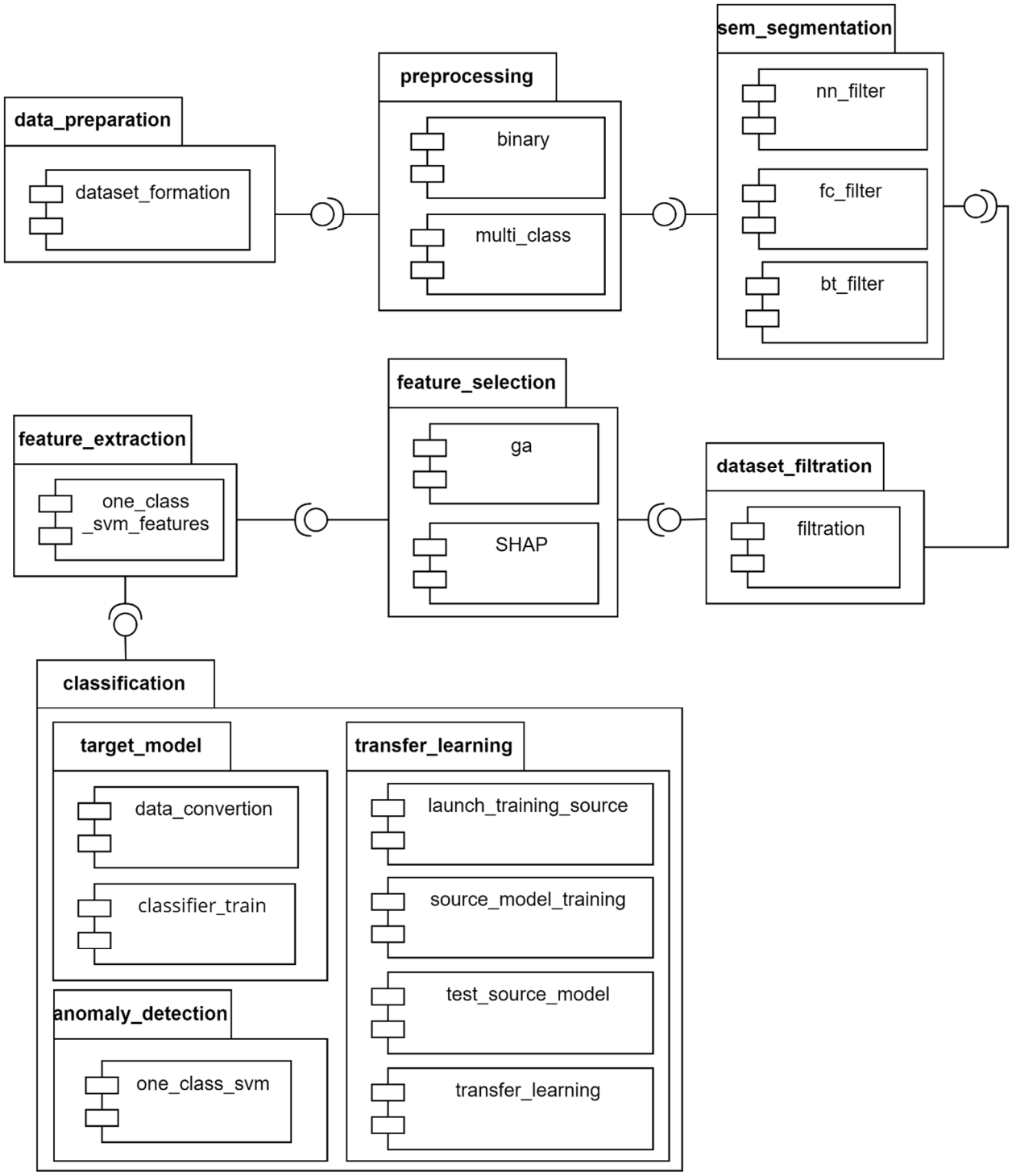


Рисунок 6.21 – UML диаграмма компонентов PyDestructiveDetector

Модуль *one_class_svm_features.py* работает с предварительно обработанным датасетом и извлекает признаки на основе ключевых слов, полученных на этапе семантической сегментации. Причем ключевые слова можно корректировать в зависимости от необходимости. В результате работы скрипта пользователь получает

датасет с признаками по каждому тексту и несколько визуализаций, отражающих корреляцию данных признаков и значения их частот в наборе.

Модуль *filtration.py* отфильтровывает строки поданного на вход датасета по условию наличия в них *n* ключевых слов из кластеров. В результате работы скрипта пользователь получает отфильтрованный датасет.

Для отбора признаков используются генетический алгоритм или метод на основе теории игр SHAP. Модуль генетического алгоритма *ga.py* работает с датасетом, сформированным в результате извлечения признаков. Код настроен на выполнение отбора признаков и оценку производительности классификатора One-Class SVM на выбранных признаках. Лучший набор признаков определяется в процессе эволюции, и его оценка приспособленности записывается на каждом поколении. В результате исполнения программы пользователю предоставляется графическая интерпретация изменения значения целевой функции в каждом поколении. В случае отбора признаков на основе *shap.py*, устанавливается важность каждого признака и его вклад в итоговый результат классификации. В результате исполнения программы пользователю предоставляется графическая интерпретация вклада каждого признака в полученных результат (рисунок 6.22).

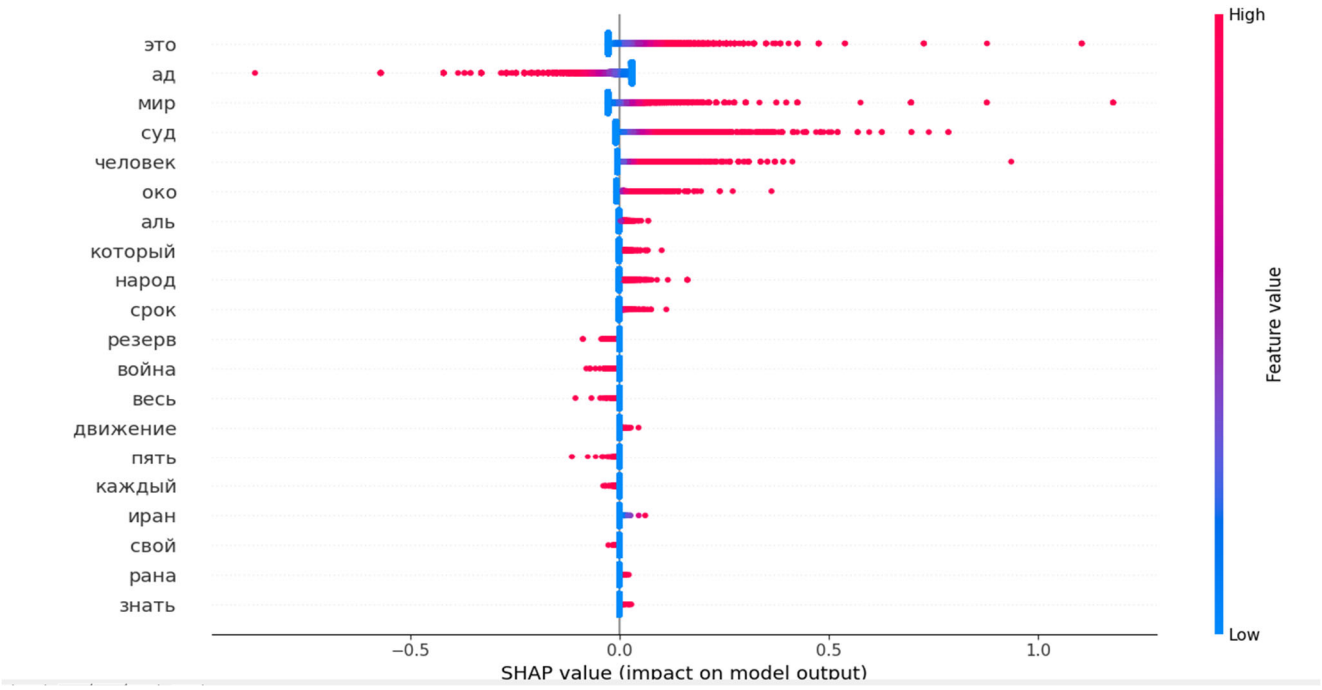


Рисунок 6.22 – Отбор признаков SHAP

Модуль *data_convertion.py* разделяет датасет на выборки для теста и обучения, приводит метки к виду, ожидаемому модулем обучения *classifier_train.py.*

Модуль *launch_training_source.py* задает необходимые параметры для обучения исходной модели (в случае если необходимо ее полное переобучение).

Модуль *source_model_training.py* представляет собой настройку модели BERT для классификации текста. Данные токенизируются с использованием токенизатора BERT, и полученные токены используются для создания наборов данных для обучения, оценки и тестирования.

Оценка исходной модели представлена в модуле *test_source_model.py*. Скрипт загружает предварительно обученную модель BERT для классификации текста на русском языке и соответствующий токенизатор и словарь, который представляет собой соответствие между числовыми идентификаторами и темами. Этот словарь используется для интерпретации результатов модели. Модель BERT используется для вычисления выхода для данного входа (идентификаторы токенов и маска).

Модуль *transfer_learning.py* отвечает за имплементацию трансферного обучения и представляет собой процесс обучения и оценки модели для классификации текстовых данных с использованием предварительно обученных эмбеддингов BERT и модели GRU+CNN. Рассчитываются и выводятся метрики точности, полноты и *F1*-меры. Псевдокод алгоритма представлен в приложении Б.

Программа способна определять автора текста с точностью до 94% в случае закрытого, и до 91% в случае открытого множества кандидатов. Определять деструктивный и экстремистский контент с точностью до 94%. Необходимо использовать не менее 50 образцов текстов каждого автора для идентификации автора, и 20000 и 59000 обучающих образцов каждого класса для определения деструктивного контента в случае иностранных агентов и экстремистов, соответственно. Примеры работы приведены на рисунке 6.23.

```
Введите свой текст без пустых строк
Россия это страна, в которой мирно уживаются большое количество национальностей. Примерное коли
чество – 160 национальностей.Каждая национальность исповедует свою религию. Поэтому в России мо
жно встретить христиан, мусульман, буддистов, иудеев и много еще различных религий.

Predicted: Не является иностранным агентом
Введите свой текст без пустых строк
```

```
Введите свой текст без пустых строк
Мы видим, что Кремль хочет запугать всех потенциальных оппонентов. Власть готова преследовать в
сех, кто хоть намеком показывает свою антивоенную позицию. В государственных СМИ говорят о необ
ходимости решительных военных действий, причем интонация там довольно брутальная. Все это контр
астирует с образом современного, мудрого и уважаемого в мировом сообществе правителя, который б
ыл так важен режиму Путина в его первые два срока и особенно во время президентства Медведева.

Predicted: Иностранный агент
Введите свой текст без пустых строк
```

Рисунок 6.23 – Примеры работы программы PyDestructiveDetector

6.7 Базы данных естественно и искусственно-языковых текстов для определения авторства

В результате диссертационной работы также разработаны и зарегистрированы БД естественно- и искусственно-языковых текстов для проведения экспериментов по идентификации авторства [307, 308, 309].

6.7.1 Наборы данных для проведения экспериментов по идентификации автора естественно-языкового текста

6.7.1.1 Набор данных художественных текстов

Корпус художественных текстов включает 1100 текстов на русском языке, созданных 100 авторами. Выбор обоснован культурной значимостью текстов и возможностью сравнения результатов с другими исследователями. Все тексты собраны из Интернет-библиотеки М. Мошкова [236]. Описание набора данных представлено в таблице 6.2.

Таблица 6.2 – Описание набора данных художественных текстов

Статистика	Значение
Количество авторов	413
Количество текстов	6132
Количество символов	332507700
Количество слов	58088948
Количество предложений	3630559
Средняя длина текста, символов	54225
Средняя длина предложения, слов	16
Среднее количество текстов каждого автора	15

6.7.1.2 Набор данных любительских текстов

Для исследования авторства любительских текстов использовались материалы онлайн-библиотеки ficbook [303]. Сайт имеет активную аудиторию и содержит множество любительских текстов («фанфиков»), написанных по мотивам известных фильмов, книг, сериалов. Все тексты сгруппированы по категориям («фандомам»).

Критериями при выборе тематических категорий для сбора служили популярность и наличие объемных текстов. Под популярностью понимается частота обновления категории, появление новых авторов и текстов. В итоге были отобраны 5 самых популярных категорий: Гарри Поттер (ГП), Вселенная Марвел (ВМ), Шерлок BBC, Наруто, Звездные войны (ЗВ).

В набор данных были включены только авторы, имеющие не менее 5 произведений по одной из категорий. Если в работе в явном виде не указан авторский вклад и не обозначено, кто автор каждого из разделов текста, то текст исключался.

Также исключались переводные тексты, поскольку при переводе авторский стиль искажается.

Сформированный набор данных (таблица 6.3) подходит для проведения моно- и кросс-тематических экспериментов. Согласно статистике собранного

набора, существенная доля авторов – создатели любительских произведений по мотивам фильмов и книг о Гарри Поттере. Однако остальные категории включают не менее 54 авторов, что позволяет формировать сбалансированные выборки текстов для исследований.

Таблица 6.3 – Информация о наборе данных любительских текстов

Характеристика набора	Все категории	ГП	ВМ	Наруто	ЗВ	Шерлок ВВС
Количество авторов	686	435	63	68	67	54
Количество текстов	6569	4138	657	663	585	526
Размер набора данных в символах	167332137	63481480	25328791	29832830	22832646	26066140
Размер набора данных в словах	40696286	23778454	5243203	4737650	3996108	2940871
Размер набора данных в предложениях	2091662	1564372	237249	188751	157949	158966
Средняя длина текста в символах	25473	103425	156742	147934	140995	155461
Максимальное количество текстов одного автора	34	21	19	22	23	22

6.7.1.3 Набор данных текстов пользователей социальных сетей

К началу 2025 года нет подходящих общедоступных текстовых наборов данных из социальных сетей для русского языка. Существует корпус, включающий коллекцию постов из Twitter* [191]. Однако этот набор данных содержит большое количество нефильтруемых рекламных материалов, вставки не только русского, но и иностранных языков. В связи с этим был собран собственный набор текстовых данных.

Комментарии пользователей были собраны с помощью API «ВКонтакте». Выбор данной социальной сети обусловлен ее популярностью среди носителей

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

русского языка. «ВКонтакте» включает пользователей разных возрастных категорий и разноплановые тематические сообщества. В таблице 6.4 представлены статистические характеристики описанного набора данных.

Таблица 6.4 – Статистические характеристики описанных наборов данных

Характеристика набора данных	Значение характеристики
Количество авторов	1114955
Количество текстов	2815104
Размер набора, символов	177683573
Размер набора, слов	27815234
Размер набора, предложений	3150806
Средняя длина текста, символов	63
Средняя длина предложения, слов	7
Среднее количество текстов на автора	3

6.7.1.4 Набор данных, полученный генеративными нейросетями

Для проведения исследований, связанных с генеративными алгоритмами, был создан дополнительный набор коротких текстов, содержащий образцы, полученные нейросетями GPT-3.5 и GPT-4 на основе базового набора комментариев «ВКонтакте». Стоит отметить, что существуют модели GigaChat [87], YandexGPT 2 [220] и YandexGPT-3 [221] и др., обученные и специализированные для русскоязычных текстов. Однако в текущей версии использование этих моделей для генерации текста в рамках решаемых в диссертационной работе задач было признано невозможным по нескольким причинам. В частности, модели не могут генерировать тексты, содержащие обсценную лексику, даже если это не нарушает законодательство РФ. Эти ограничения могут быть обусловлены этическими соображениями, заложенными разработчиками, стремлением избежать распространения нежелательного контента и улучшить безопасность взаимодействия с моделями. В некоторых случаях при попытке генерации текста модели возвращали образцы,

идентичные оригиналам, что также указывает на их ограниченность. По указанным причинам было принято решение отказаться от работы с русскоязычными моделями GigaChat и YandexGPT 2, 3 в их текущей версии.

Для того, чтобы сгенерированные тексты были пригодны для дальнейшей работы, на процесс генерации накладывались следующие ограничения.

1. Генерировались тексты длиной 50 и более символов.

2. Использовались только тексты на русском языке.

3. Сгенерированные тексты должны включать те же эмодзи, что и оригинальные сообщения.

4. Модель должна сохранять оригинальный стиль письма, включая пунктуационные, орфографические и семантические ошибки и особенности.

Информация о полученном наборе данных приведена в таблице 6.5.

Таблица 6.5 – Статистические характеристики полученного набора данных

Характеристика	GPT-3.5	GPT-4
Количество авторов	100	100
Количество текстов	5742	5000
Количество символов	655690	559640
Количество слов	27133	53380
Количество предложений	4448	5392
Средняя длина текста, символов	114,2	111,9
Средняя длина предложения, слов	6,1	9,9
Среднее количество текстов каждого автора	57	50

Такой набор данных можно использовать для экспериментов по определению авторства только на сгенерированных образцах, определению авторства в случае парного сравнения оригинального автора и сгенерированного на его основе «клона», а также путем внедрения искусственных образцов на этапе тестирования для решения задачи открытой атрибуции.

Необходимо отметить, что при использовании сгенерированных текстов для определения авторства важно обеспечить их соответствие реальному авторскому стилю. Одной из часто используемых мер подобия для текста является косинусное

сходство (см. формулу (2.38)). Косинусное сходство измеряет угол между векторами представления текстов и может служить показателем их семантической близости. Выбор данной метрики обусловлен ее применением в смежных работах, посвященных анализу естественного языка [8, 121].

Чтобы сравнить исходные и сгенерированные сообщения по идентификатору пользователя, были выполнены следующие действия:

1. Разделение исходных и сгенерированных сообщений на два разных датафрейма на основе префикса идентификатора пользователя.

2. Для каждого идентификатора пользователя создан список исходных и сгенерированных сообщений.

3. Проведено преобразование каждого сообщения в матрицу TF-IDF.

4. Вычислено косинусное сходство между каждой парой исходных и сгенерированных сообщений.

5. Рассчитано среднее косинусное сходство для всех пар сообщений для каждого идентификатора пользователя, что дает меру того, насколько похожи сообщения в целом для каждого пользователя.

Для более полной оценки качества генерации текста проведен анализ структуры предложений в оригинальных и сгенерированных текстах при помощи метрик BLEU (Bilingual Evaluation Understudy) (6.1) или ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (6.2):

$$BLEU = min\left(1, \frac{output\_length}{reference\_length}\right)\left(\prod_{i=1}^{n} precision_i\right)^{\frac{1}{n}}, \tag{6.1}$$

$$ROUGE = \frac{count_{match}\left(gram_n\right)}{count\left(gram_n\right)}, \tag{6.2}$$

где *output_length* и *reference_length* – длины исходного и сгенерированного образцов; $count_{match}\left(gram_n\right)$ – количество *n*-грамм в исходном образе; $count\left(gram_n\right)$ – количество *n*-грамм в сгенерированном образце.

Метрики BLEU и ROUGE являются мерами сходства между текстами на основе перекрытия слов или фраз. Они оценивают грамматическую правильность,

структурную целостность и плавность выражения в тексте. Значение метрик равное 1 говорит, что тексты идентичны, 0 – тексты не имеют сходства.

Метрики считались для каждого автора для его оригинальных и сгенерированных образцов. Усредненное значение по всем авторам представлено в таблице 6.6.

Таблица 6.6 – Оценка сходства оригинальных и сгенерированных образцов

Модель	Косинусное сходство	BLEU	ROUGE
GPT-3.5	0,7	0,58	0,61
GPT-4	0,43	0,35	0,41

Согласно полученным значениям, можно отметить высокое сходство текстов, созданных GPT-3.5 с исходными образцами. Но если рассматривать метрики сходства не только как показатели точного соответствия оригинальным текстам, но и как индикаторы уникальности и креативности генерации, то более низкие значения метрик для GPT-4 могут свидетельствовать о ее способности создавать тексты, которые не являются простыми копиями оригинала, а представляют собой более глубокую и осмысленную обработку исходной информации.

6.7.2 Набор данных для проведения экспериментов по идентификации автора исходного кода

Для качественного обучения и оценки модели идентификации автора исходного кода программ необходим объемный набор данных. С целью получения такого набора рассматривались крупнейшие хостинги IT-проектов GitHub [304] и GitLab [89]. Оба хостинга предоставляют открытый доступ к исходным кодам проектов на множестве языков программирования. Отбор выполнялся только из репозиториев, отмеченных «звездами» (т.е. популярных программистов), а также из репозиториев, авторы которых имели высокую частоту коммитов. Сами коммиты

должны были быть подтверждены со стороны GitHub и иметь статус «Verified». Авторы, исходные коды которых вошли в набор данных, проходили итерационную проверку научной группой Романова А.С.

Сбору подлежали исходные коды программ на самых популярных языках программирования: Python, Java, C++, C#, JavaScript, PHP, Kotlin, Swift, Go, Ruby, C, Groovy, Perl. Соотношение числа проектов и отдельных файлов для каждого языка программирования представлено на рисунке 6.23. Подробная информация о сформированном наборе (таблица 6.7).

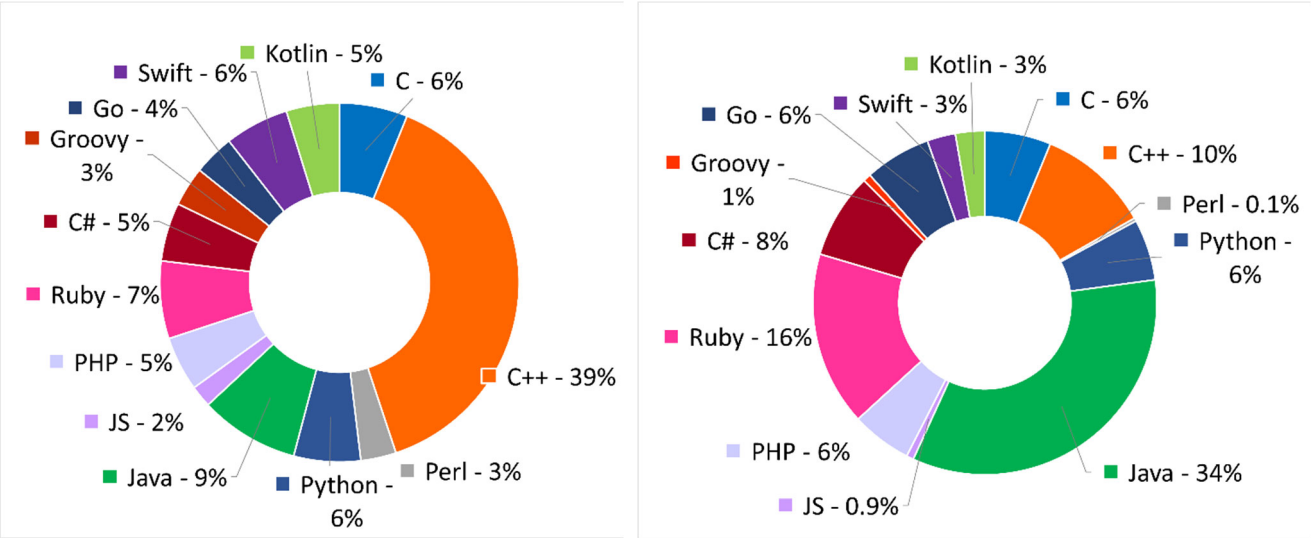


Рисунок 6.23 – Соотношение числа проектов (слева) и файлов с кодом (справа) по языкам

Таблица 6.7 – Статистика набора данных

Количество строк кода	20967040
Количество проектов	569
Количество проектов с одним автором	71
Средняя длина коммита, строк	13
Максимальная длина коммита, строк	119892
Среднее число исходных кодов в проекте	231
Средняя длина исходного кода, строк	169
Суммарное число символов кода	212336673
Всего исходных кодов	102908
Авторов во всех проектах	383

6.7.3 Набор данных для проведения экспериментов по определению возраста

Для формирования набора данных была выбрана социальная сеть «ВКонтакте». Использовался набор из пп. 6.7.1.3, в котором дополнительно была собрана и (или) добавлена информация о возрасте пользователя. В этой социальной сети достаточно сообщений взрослых, подростков и детей, темы обсуждениях которых могут быть очень разнообразны. Представители младших возрастных групп могут обсуждать школу, хобби и кумиров в то время, как обсуждения старшей возрастной группы могут строиться вокруг работы, быта, здоровья и т.п. Часть тем может быть интересна представителям всех возрастных групп.

В таблице 6.8 представлена информация о собранном наборе данных с распределением по возрастным группам. В таблице 6.9 та же информация представлена в соответствии с классификацией ВОЗ.

Таблица 6.8 – Информация о наборе данных

Количество несовершеннолетних пользователей	18283
Количество пользователей от 18 до 21 года	193348
Количество пользователей от 21 до 24 лет	189299
Количество пользователей от 24 до 27 лет	199387
Количество пользователей от 27 до 30 лет	175265
Количество пользователей от 30 до 35 лет	86989
Количество пользователей от 35 до 45 лет	73086
Количество пользователей старше 45 лет	179298

Таблица 6.9 – Информация о наборе данных согласно ВОЗ

Количество пользователей от 18 до 44 лет	917374
Количество пользователей от 45 до 59 лет	74070
Количество пользователей от 60 до 74 лет	59338
Количество пользователей от 75 до 90 лет	13036
Количество пользователей старше 90 лет	32854

Сформированный набор данных охватывает все представленные в социальной сети возрастные категории и слои общества. Большинство текстов в наборе имеют объем до 300 символов и содержат большое количество знаков пунктуации, эмотиконов и авторских неологизмов, грамматических и орфографических ошибок.

6.7.4 Наборы данных для проведения экспериментов по определению деструктивной и экстремисткой направленности текста

Сбор данных осуществлялся посредством API социальной сети «ВКонтакте» и мессенджера Telegram и включал предварительную идентификацию деструктивных и новостных сообществ и каналов, и сбор комментариев пользователей из них.

В противовес были собраны тексты, не имеющие деструктивной и экстремистской направленности (новостные каналы, религиозные, знаменитостей и т.д.).

Статистические характеристики собранных наборов данных приведены в таблице 6.10.

Таблица 6.10 – Статистические характеристики датасетов

Статистика	Telegram экстремисты	Telegram иноагенты	Telegram новости
Количество авторов	1844	1921	101163
Количество текстов	25336	33028	437532
Количество символов	678432	1051176	12688428
Количество слов	136380	264737	2114738
Количество предложений	27275	52946	352456
Средняя длина текста, символов	27	32	29

Собранный датасет можно использовать в экспериментах, предназначенных для разделения авторства между комментариями обычных граждан и текстов де-

структивной или экстремистской направленности, определения авторства исключительно на деструктивных образцах, а также путем внедрения деструктивных или экстремистских образцов на этапе тестирования для решения задачи открытой атрибуции.

В исследованиях также использовались материалы с соревновательной платформы Kaggle. Первый набор данных «How ISIS Uses Twitter» [103] включает сообщения из социальной сети Twitter*, опубликованных последователями экстремистского движения ИГИЛ** в процессе подготовки и осуществления терактов.

Второй набор данных «Religious Texts Used By ISIS» [183] содержит короткие тексты из пропагандистских журналов «Дабик»*** и «Румийя»***, предназначенных для вербовки сторонников террористической организацией ИГИЛ**.

Оба датасета были переведены на русский язык с помощью сервиса deep-translator [61] на базе переводчика от компании Google.

Также использовался датасет комментариев пользователей форума KavkazChat*** [59], признанного экстремистским. Форум был организован по тематическим разделам, включающим стандартные обсуждения домашнего быта, религии, хобби, спорта. В качестве экстремистских же использовались только разделы, содержащие запрещенные тематики.

Информация о сторонних датасетах приведена в таблице 6.11.

Таблица 6.11 – Статистические характеристики датасетов

Статистика	How ISIS Uses Twitter	Religious Texts Used By ISIS	KavkazChat***
Количество авторов	113	1160	7125
Количество текстов	17411	2641	699981
Количество символов	1889126	1108404	126136576
Количество слов	272700	196941	9294114
Количество предложений	23184	8139	1388431
Средняя длина текста, символов	108	419	180

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.
** Запрещенная в Российской Федерации террористическая организация.
*** Признаны экстремистскими в Российской Федерации.

6.7.5 Набор данных для экспериментов по определению пола и гендера

Сбор данных для первого набора, содержащего сообщения представителей ЛГБТ-сообщества*, выполнен посредством осуществления запросов к API мессенджера Telegram. Telegram служит платформой для широкой аудитории, включая мужчин и женщин, ведущих обсуждения на разнообразные темы, а также представителей ЛГБТ-сообщества*, которые делятся своим опытом и взглядами, не всегда прямо связанными с их гендерной идентичностью или сексуальной ориентацией.

Был осуществлен сбор новостных публикаций в каналах, которые ведут мужчины и женщины. Всего было задействовано 127 каналов. Подробная информация о наборах данных новостей из каналов мужчин и женщин, а также представителей ЛГБТ* приведена в таблицах 6.12, 6.13.

Таблица 6.12 – Набор данных новостных публикаций мужчин и женщин (telegram)

Характеристика	Мужчины	Женщины
Количество текстов	48491	76034
Количество символов	28532043	28759251
Количество слов	4138153	4417450
Количество уникальных слов	608277	562910
Число предложений	34333	353729
Число каналов	25	101
Среднее число сообщений в канале	1940	753

Таблица 6.13 – Набор данных новостных публикаций ЛГБТ* (telegram)

Характеристика	Геи	Лесбиянки	Бисексуалы	Трансексуалы	Общие каналы ЛГБТ*
Количество текстов	1681	2527	3192	5381	26272
Количество символов	1105241	2873749	2473842	3235497	3007826
Количество слов	145191	402696	377448	528323	428869
Количество предложений	13880	35429	31059	38891	33232
Количество каналов	9	5	5	10	11
Среднее количество сообщений в канале	187	505	638	538	2388

* Запрещенная в Российской Федерации экстремистская организация.

Акцент был сделан также на формировании набора данных комментариев, поскольку комментарии часто представляют собой неподготовленные реакции пользователей, отражающие их искренние мысли и чувства в данный момент. В отличие от контента, создаваемого блогерами и владельцами коммерческих каналов, для которого зачастую привлекаются профессиональные СММ-специалисты с целью создания привлекательного и подготовленного материала, комментарии остаются непосредственным выражением мнений и взглядов обычных пользователей, что делает их ценным источником для анализа естественного языка.

Поскольку Telegram не предоставляет данных о поле пользователя, был реализован собственный механизм автоматической разметки пола на основе доступных данных об аккаунте (имя, фамилия, ник пользователя) и словаря русских имён и фамилий.

Информация о размеченных наборах данных комментариев после фильтрации приведена в таблицах 6.14, 6.15.

Таблица 6.14 – Набор данных мужских и женских комментариев (telegram)

Характеристика	Мужчины	Женщины
Количество текстов	60354	63397
Количество символов	9939081	9494575
Количество слов	1580641	1515193
Количество предложений	132865	142274
Количество пользователей	8497	14298
Среднее количество сообщений пользователя	7	4

Таблица 6.15 – Набор данных комментариев ЛГБТ* (telegram)

Характеристика	Геи	Лесбиянки	Бисексуалы	Транссексуалы	Общие каналы ЛГБТ*
Количество текстов	42594	20212	2913	5935	96486
Количество символов	6748182	4974809	749093	1003100	17785887
Количество слов	1085463	801562	118748	161875	2857963
Количество предложений	86642	55411	9448	11694	216405
Количество пользователей	3429	1998	655	1009	13253

Для формирования второго набора данных была выбрана социальная сеть X* (бывший Twitter*) поскольку она не блокирует контент, связанный с тематикой ЛГБТ**. Социальная сеть предоставляет богатый источник информации для анализа как сообщений представителей ЛГБТ-сообщества**, так и гетеросексуальных пользователей. Для составления датасета был проведен ручной поиск по ключевым словам, нацеленный на выявление аккаунтов, активно обсуждающих тематику ЛГБТ**. По результатам поиска был сформирован список идентификаторов пользователей, представляющих тот или иной гендер. Рассматривались только русскоязычные пользователи. Из комментариев к сообщениям были выделены следующие группы идентификаторов, которые также проверялись по ключевым словам и вручную и т.д. После составления итогового списка идентификаторов, использовалась специально разработанная программа для автоматизированного сбора сообщений. Информация об итоговых датасетах сообщений мужчин, женщин, и представителей ЛГБТ-сообщества** представлена в таблицах 6.16 и 6.17.

Полученные наборы данных являются уникальными, поскольку к началу 2025 года не опубликовано научных статей и наборов данных, результаты которых основаны на представительных корпусах, содержащие тексты представителей ЛГБТ-сообщества** на русском языке.

Таблица 6.16 – Набор данных публикаций ЛГБТ (X*)

Характеристика	Геи	Лесбиянки	Бисексуалы	Трансексуалы	Общие сообщения ЛГБТ**
Количество текстов	31782	36857	30061	30931	33038
Количество символов	3988324	4486219	3868860	4704609	4567291
Количество слов	621212	715084	605769	728612	692035
Количество предложений	49166	36824	43998	45135	42 278
Количество пользователей	81	65	106	80	63
Среднее число сообщений пользователя	392	567	284	387	525

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.
** Запрещенная в Российской Федерации экстремистская организация.

Таблица 6.17 – Набор данных мужских и женских публикаций (X*)

Характеристика	Мужчины	Женщины
Количество текстов	44470	43570
Количество символов	5932698	5377996
Количество слов	891616	704138
Количество предложений	71959	68833
Количество пользователей	69	64
Среднее количество сообщений пользователя	644	680

6.7.6 Набор данных для экспериментов по проверке однородности и поиску плагиата

При составлении корпуса особое внимание уделяется текстам, создаваемым в образовательной и академической среде. К таким текстам относятся научные статьи, технические отчеты, дипломные работы, диссертации, сочинения, эссе и т.д. В качестве экспериментальных данных было решено рассмотреть находящиеся в открытом доступе тексты, создаваемые молодыми учеными, а также соискателями ученых степеней: диссертационные работы и научные статьи на русском языке (таблица 6.18).

Таблица 6.18 – Статистика по набору данных научных текстов

Характеристика	Научные статьи	Диссертации	GPT-3.5	GPT-4
Количество авторов	3971	525	250	250
Количество текстов	4503	525	1000	1000
Ср. длина, слов	2713	31762	1521	1645
Ср. длина, символов	21401	247743	12670	13688
Ср. длина, предложений	208	2443	163	180
Статей с 1 автором	281	–	–	–
Статей с 2-5 авторами	2317	–	–	–
Статей с >5 авторами	1902	–	–	–

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

Данные были собраны из открытых источников, содержащих журналы, рецензируемые ВАК по техническим, физико-математическим, химическим, медицинским и филологическим наукам (рисунок 6.24). Большая часть текстов была получена из таких изданий, как «Вестник РАМН», «Вестник НГУ», «Вестник ТГУ», «Вестник МГУ», «Вестник СПбГУ», «Труды СПИИРАН», «Доклады ТУСУР». В состав набора были также включены 40 научных статей исследовательской группы А.С. Романова с заранее известными фрагментами, написанными соавторами.

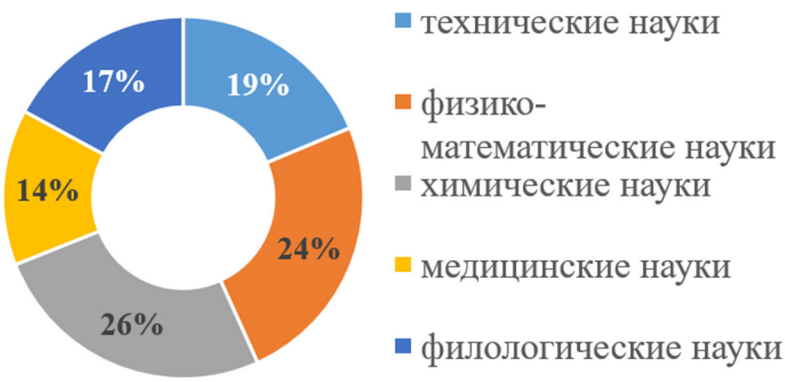


Рисунок 6.24 – Распределение статей по тематикам

На основе собранного датасета были также сгенерированы тексты по каждой тематике с помощью генеративных моделей GPT-3.5 и GPT-4. Процесс генерации был аналогичен описанному в пп. 6.7.1.4.

6.7.7 Достоверность данных

Проблема достоверности авторских атрибутов для художественных текстов решалась за счет использования авторов, обладающих значительным объемом литературного творчества. Все случаи подобные соавторству, использованию псевдонимов и др. известны заранее, поэтому такие тексты в исследованиях не использовались.

Для коротких текстов социальной сети «ВКонтакте» использовалось несколько подходов. Во-первых, на начальном этапе проводилась автоматическая

фильтрация текстов – использовались только авторы с непустыми профилями, у которых есть несколько фотографий, указан возраст, пол. Дополнительно проводилась фильтрация рекламных ботов. Все данные, вызывающие сомнения, в датасет не включались. Во-вторых, использовались только пользователи «с галочками», обозначающими, что профиль пользователя подтвержден через Госуслуги или банковские сервисы (подтверждается фамилия и имя). По имени и фамилии с использованием словарей производилась разметка пола. Для пользователей, имена которых не были найдены в словаре, производилась дополнительная ручная проверка аккаунта и уточнение половой принадлежности или они исключались. Определение возраста пользователей представляло собой более сложную задачу, поскольку во «ВКонтакте» возможно произвольное изменение информации о возрасте, которая не подлежит верификации. Для решения данной проблемы использовалась нейросеть для сопоставления указанного возраста с возрастом, определенным по фотографиям в профиле. Если разница между этими значениями не превышала двух лет, возраст считался установленным корректно, и тексты таких пользователей допускались к обучению моделей.

Для текстов мессенджера Telegram достоверность данных обеспечивалась за счет того, что список источников и авторов формировался согласно задаче.

Достаточным основанием для того, чтобы считать канал «экстремистским», считается выполнение каналом, на котором опубликован текст, как минимум одного из следующих условий:

а) владелец канала включен в «Перечень террористов и экстремистов», размещенный на портале Росфинмониторинга (Федеральная служба по финансовому мониторингу РФ) [280]; и (или)

б) характер публикуемых на канале материалов носит признаки экстремистского и (или) террористического содержания согласно положениям Федерального закона от 25 июля 2002 г. N 114-ФЗ «О противодействии экстремистской деятельности» (с изменениями и дополнениями) [327]:

– публичные призывы насильственно изменить конституционный строй РФ, вывести из ее состава город или регион, препятствовать сотрудникам госорганов выполнять свою работу, в том числе посредством применения к ним насилия;

– публичное оправдание терроризма; организация и координация вышеупомянутой деятельности, а также подстрекательство к ней и распространение способствующих ее осуществлению материалов;

– финансирование вышеупомянутой деятельности и оказание содействия ее организаторам и исполнителям.

В случае иностранных агентов, достаточным основанием считается выполнение каналом, на котором опубликован текст, как минимум одного из следующих условий:

а) физическое лицо или организация, которой принадлежит канал, имеет статус иностранного агента согласно «Реестру иностранных агентов» [282], сформированного министерством юстиции Российской Федерации; и (или)

б) ресурс включен в «Реестр информационных ресурсов иностранных агентов, доступ к которым ограничен на территории Российской Федерации» [283]; и (или)

в) характер публикуемых на канале материалов носит признаки иностранного агента согласно положениям Федерального закона от 14.07.2022 № 255-ФЗ «О контроле за деятельностью лиц, находящихся под иностранным влиянием» [325]:

– организация/физическое лицо, получающее поддержку и (или) находящееся под иностранным влиянием и осуществляющее политическую деятельность, целенаправленный сбор сведений в области военной, военно-технической деятельности Российской Федерации, распространение сообщений и материалов и (или) участие в создании таких сообщений и материалов;

– организация/физическое лицо, в ходе деятельности проводящие атаки на структуры гражданского общества с целью их дискредитации и создания юридических предпосылок для их преследования;

– организация/физическое лицо, распространяющее недостоверную информацию о принимаемых органами публичной власти Российской Федерации решениях и проводимой ими политике;

– организация/физическое лицо, участвующее в деятельности организации, включенной в перечень иностранных и международных неправительственных организаций, деятельность которых признана нежелательной на территории Российской Федерации.

Помимо проверок на соответствие перечням, применительно к пользователям, состоящим в канале, выполнялся дополнительный этап фильтрации за счет семантической кластеризации, направленной на отнесение их постов к тематикам, релевантным для представителей ЛГБТ-сообщества, экстремистов или иноагентов. ЛГБТ тоже относится к экстремистским организациям. На этапе фильтрации отбирались только те аккаунты, владельцы которых открыто заявляют о своей причастности к ЛГБТ-сообществу.

В случаях, когда автоматизированная проверка вызывала сомнения, на различных этапах исследования привлекались филологи – преподаватели и студенты старших курсов Томского государственного университета и Высшей школы экономики для очистки данных от недостоверной информации.

В случае с чувствительными данными, проверка которых осложняется этическими и правовыми факторами, не исключены ошибки при фильтрации. Однако небольшая доля недостоверных данных как правило не приводит к катастрофическим последствиям при условии достаточного количества данных и их качественной подготовки. «Мусорные» данные иногда специально добавляют в датасеты, чтобы повысить устойчивость получаемых с их помощью моделей к ошибкам.

При сборе исходных кодов программ часть данных были получены из достоверных источников – датасетов соревнований, участники которых доподлинно известны. Остальные данные были получены с платформы GitHub, для которых отбор выполнялся только из репозиториев, отмеченных звездами (т.е. популярных программистов), а также из репозиториев, авторы которых имели высокую частоту коммитов. Сами коммиты должны были быть подтверждены со стороны GitHub и

иметь статус «Verified». Авторы, исходные коды которых вошли в набор данных, проходили итерационную проверку научной группой Романова А.С.

6.8 Выводы по главе

1. Разработан оригинальный программный комплекс для выделения информативных признаков на основе семантического анализа текста «PyAuthorship». В рамках которого реализованы подсистемы сбора образцов текстов, хранения информации, анализа естественного и искусственного языка, а также выделения информативных признаков на основе семантического анализа. Представлено описание каждой из разработанных подсистем, оригинального алгоритмического обеспечения, на основе которого они функционируют, и интерфейса взаимодействия с ними. На основе наиболее эффективных семантических методов извлечения информативных признаков комплекс определяет подмножество признаков, которое используется для исследований в рамках решения прикладных задач кибербезопасности.

2. Разработано оригинальное алгоритмическое и программное обеспечение программной системы «Авторовед», позволяющее идентифицировать автора из потенциально возможных с помощью нейронных сетей и машины опорных векторов, выделять заимствованный текст, подтверждать или опровергать авторство текста.

3. Разработана программа прикладная программа для определения возраста автора текстового сообщения в социальных сетях и мессенджерах «Age Detector».

4. Разработано оригинальное алгоритмическое и программное обеспечение «CoDEtective», учитывающее в том числе сложные случаи, когда код написан в соавторстве, применяются обфускаторы и др., для идентификации автора программного кода, написанного на 13 наиболее популярных языках программирования.

5. Разработана прикладная программа для определения эмоциональной окраски текста «Moodorrude».

6. Разработаны оригинальные программные комплексы «PyDestructiveDetector» и «ProGender», а также алгоритмическое обеспечение, предназначенное для идентификации текстов, содержащих пропаганду нетрадиционных ценностей, деструктивную и экстремистскую направленность, потенциально нарушающих законодательство РФ. А также идентификации лиц, распространяющих подобные текстовые материалы.

7. Разработана структура баз данных для хранения текстов, созданных в результате повседневной, профессиональной и творческой деятельности автора, учитывающая особенности каждой группы текстов. Собраны объемные и релевантные датасеты, предназначенные для решения частных практических задач кибербезопасности.

Свидетельства о регистрации программ для ЭВМ и баз данных представлены в приложении В.

ЗАКЛЮЧЕНИЕ

Исследования, выполненные в диссертационной работе, позволили решить **научную проблему** – создание комплексной методологии идентификации автора текстовой информации, **имеющую важное хозяйственное значение, внедрение которой вносит значительный вклад** в развитие кибербезопасности, защиты авторского права, компьютерной лингвистики. Разработанная методология и инструментарий на ее основе может использоваться для защиты интеллектуальной собственности, идентификации автора сообщений в сети Интернет и продленной аутентификации пользователей социальных сетей, выявления признаков пропаганды ЛГБТ* и педофилии, определения автора-вирусописателя, определения эмоциональной окраски текста, деструктивных, а также текстов экстремистской направленности, запрещенных законодательством Российской Федерации.

Основные результаты диссертационной работы.

1. Проведен подробный анализ основных работ в области идентификации автора естественного и искусственного текста, методов профилирования автора на основе его профессиональной деятельности, возрастных и гендерных атрибутов, идеологии, тональности публикуемого текста. Рассмотрены работы, направленные на атаки методов путем внедрения образцов, созданными генеративными моделями, методов имитации и обфускации текстового материала.

2. Разработана комплексная методология идентификации автора текстовой информации для решения задач кибербезопасности, включающая широкий спектр методов, моделей и алгоритмов, основанных на анализе как естественно-языковых, так и искусственно-языковых текстов, предоставляющая универсальный инструментарий для классификации текстов. Методология предусматривает применение методов статистического анализа, машинного и глубокого обучения, что делает её адаптируемой к различным задачам и способной эффективно бороться с потенциальными атаками.

---

* Запрещенная в Российской Федерации экстремистская организация.

3. Предложена модель, учитывающая как внутренние компоненты процесса создания автором текста в киберсреде, так и внешние ограничения, накладываемые личностью автора, видом деятельности, в том числе профессиональной и досуговой, особенностями среды и семантики текста, что позволяет не только точно идентифицировать автора, но и учитывать специфику его взаимодействия с киберпространством.

4. Предложена новая методика идентификации автора русскоязычного текста. Особое внимание уделено методам отбора информативных признаков в сочетании с SVM, среди которых генетический алгоритм и регуляризационный отбор были признаны наиболее эффективными для классификации художественных текстов, для коротких текстов выбрана гибридная модель на основе комбинации GRU+CNN. Эксперименты показывают, что предложенная методика обеспечивает высокую точность классификации (до 99% для 2 и 5 авторов, 94% для 10 и 88% для 20), сопоставимую или превосходящую аналоги. Было установлено, что методика на основе одноклассового SVM и GRU+CNN обеспечивает высокую точность (от 98% до 75% для 3–20 классов) при работе с открытым множеством кандидатов. Методика применима к задаче верификации автора текста и способна решать ее с точностью 99,2%, 98,9% и 97,8% для художественных, любительских и коротких текстов, соответственно.

5. Разработана методика определения деструктивной и экстремистской направленности в текстах, дополняющая базовую методику идентификации автора естественного текста за счет интеграции семантической кластеризации текста и трансферного обучения. Установлено, что добавление трансферного обучения позволяет получить прирост точности до 10% и 12% в случае идентификации автора и определения деструктивного контента, соответственно, а добавление семантической кластеризации – 9% для тех же случаев. Совместное использование инструментов дает совокупный прирост точности 14% и 20%, соответственно.

6. Разработана методика идентификации пола и гендера автора текстовой информации, впервые учитывающая особенности, характерные для текстов,

созданных представителями ЛГБТ-сообщества*. Методика основана на принятии решений ансамблем SVM, CNN и BERT. Признаки для SVM сформированы на основе частотных распределений ключевых слов, полученных с помощью семантической кластеризации, а признаки для CNN – на основе сглаженных вероятностей триграмм символов. Ансамбль при разграничении по половому признаку (мужчины и женщины) достигает точности 92%, точность разграничения по гендеру – до 93%. Для наиболее сложного случая классификации шести гендерных групп точность составляет 68%.

7. Предложена методика определения возраста автора текста, основанная на модели fastText для анализа текстов и методе фильтрации пользовательских фотографий с помощью модели VGG-Face. Определение возраста по фотографии через VGG-Face и последующее сравнение полученных данных с возрастом, указанным в профиле пользователя, обеспечивает верификацию корректности пользовательских данных. Процедура фильтрации оказывает значительное влияние на точность распределения возрастных данных и качество итоговой модели. Для двух возрастных категорий (младше 18 лет и старше 21 года) точность составила 82%, а для трех категорий (младше 18, от 21 до 27 и старше 30 лет) – 63%.

8. Разработаны репрезентативные наборы данных для обучения и оценки методического и алгоритмического обеспечения. Набор данных для идентификации авторов естественно-языковых текстов в целях решения задач кибербезопасности включают художественные, любительские, научные тексты, тексты пользователей социальных сетей, а также искусственные тексты, созданные генеративными моделями на их основе. Набор данных для идентификации авторов искусственно-языковых текстов включают исходные коды программ на 13 языках программирования; обфусцированные исходные коды на 5 языках программирования; исходные коды авторов, владеющих 2 и более языками программирования; коммиты исходных кодов программ, написанных группой разработчиков; исходные коды, написанные по стандартам Linux Kernel; исходные коды, сгенерированные моделями семейства GPT.

---

* Запрещенная в Российской Федерации экстремистская организация.

9. Разработан классификатор на основе модели CodeBERT и методика идентификации автора исходного кода программы на его основе, учитывающая простые (точность до 93%) и сложные (точность до 85%) прикладные случаи. Полученные результаты подтверждают независимость методики от языка программирования автора, а также ее устойчивость к преднамеренным преобразованиям и следованию стандартам кодирования.

10. Разработана методика проверки однородности текста и поиска заимствований. Методика основана на комбинации сиамской сети SimNN, обученной с контрастивной и тройной функциями потерь, для подтверждения автора двух и более текстов/фрагментов, а также подхода к определению авторства неоднородных фрагментов. Точность методики для выявления неоднородных фрагментов составляет 94%, для определения авторства фрагментов – более 90%, для подтверждения автора – до 99%.

11. На основе разработанной методологии и методик создано алгоритмическое и программное обеспечение для анализа естественно- и искусственно-языковых текстов в задачах кибербезопасности. В его состав входят: программный комплекс для получения информативных признаков на основе семантического анализа «PyAuthorship», программная система для идентификации автора письменной речи «Автороведъ», чат-бот для определения возраста автора текстового сообщения «Age Detector», система для идентификации автора исходного кода программы «CoDEtective», программа для тонального анализа текста «Moodorrude», программный комплекс для идентификации запрещенных законодательством РФ и деструктивных текстов «PyDestructiveDetector» и программный комплекс для определения пола и гендера автора естественно-языкового текста «ProGender».

СПИСОК ЛИТЕРАТУРЫ

1. Abbasi A., Chen H. Applying authorship analysis to extremist-group web forum messages // IEEE Intelligent Systems. 2005. Vol. 20, no. 5. P. 67–75.

2. Authorship identification using ensemble learning / A. Abbasi, A.R. Javed, F. Iqbal [et al.] // Scientific Reports. 2022. no. 12. 9537 p.

3. Terrorism Attack Classification Using Machine Learning: The Effectiveness of Using Textual Features Extracted from GTD Dataset / M. Abdalsalam, C. Li, A. Dahou, N. Kryvinska // CMES-Computer Modeling in Engineering & Sciences. – 2024. Vol. 138, No. 2. URL: https://cdn.techscience.cn/files/CMES/2024/TSP_CMES-138-2/TSP_CMES_ 29911 /TSP_CMES_29911.pdf (дата обращения: 30.12.2024).

4. A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem / R. Abu Khurma, I. Aljarah, A. Sharieh, [et al.] // Mathematics. 2022. Vol. 10, no. 464.

5. SHIELD: Thwarting Code Authorship Attribution / M. Abuhamad, C. Jung, D. Mohaisen, D. Nyang. URL: https://arxiv.org/abs/2304.13255 (дата обращения: 30.12.2023).

6. Acharya S.K. Developing and Testing a Tool to Classify Sentiment Analysis. URL: https://papers.ssrn.com/ sol3/papers.cfm?abstract_id=3753005 (дата обращения: 30.12.2023).

7. Afroz S., Greenstadt R., Brennan M. Detecting hoaxes, frauds, and deception in writing style online // 2012 IEEE Symposium on Security and Privacy. IEEE, 2012. P. 461–475.

8. Akimushkin C., Amancio D.R., Oliveira O.N. On the role of words in the network structure of texts: Application to authorship attribution // Physica A: Statistical Mechanics and its Applications. 2018. Vol. 495. P. 49–58.

9. Al Medawer A. Textual Analysis and Detection of AI-Generated Academic Texts. Mid Sweden University. 2023. URL: https://miun.diva-portal.org/smash/get/diva2:1773407/FULLTEXT01.pdf (дата обращения: 30.12.2023).

10. Alowibdi J.S. Gender Prediction of Generated Tweets Using Generative AI // Information. 2024. Vol. 15. Iss. 8. 452. URL: https://www.mdpi.com/2078-2489/15/8/452 (дата обращения: 30.12.2024).

11. AlphaCode URL: https://deepmind.com/blog/article/Competitive-programming-with-AlphaCode (дата обращения: 20.02.2024).

12. Source Code Authorship Attribution using Long Short-Term Memory Based Networks / B. Alsulami, E. Dauber, R. Harang [et al.] // In Proceedings of the 22nd European Symposium on Research in Computer Security. 2017. P. 65–82.

13. Alterkavi S., Erbay H. Novel authorship verification model for social media accounts compromised by a human // Multimedia Tools and Applications. 2021. Vol. 80. P. 13575–13591.

14. Álvarez-Fidalgo D., Ortin F. CLAVE: A deep learning model for source code authorship verification with contrastive learning and transformer encoders // Information Processing & Management. 2025. Vol. 62, no. 3. 104005 p.

15. AnalyseC URL: https://github.com/ryarnyah/AnalyseC (дата обращения: 12.01.2021).

16. Program Obfuscation: A Quantitative Approach / B. Anckaert, M. Madou, B. Sutter [et al.] // In Proceedings of the 2007 ACM Workshop on Quality of Protection (QoP 2007). 2007. P. 15–20.

17. Arora M., Kansal V. Character level embedding with deep convolutional neural network for text normalization of unstructured data for Twitter sentiment analysis // Social Network Analysis and Mining. 2019. Vol. 9, no. 1. P. 1–14.

18. Attention Is All You Need / A. Vaswani [et al.] // arXiv preprint. 2017. URL: https://arxiv.org/abs/1706.03762 (дата обращения: 29.12.23).

19. Author Profiling Using Semantic and Syntactic Features Notebook for PAN at CLEF 2019 / G. Kovács [et al.] // Conference and Labs of the Evaluation Forum. 2019. URL: https://www.researchgate.net/publication/350836509_Author_Profiling_Using_Semantic_and_Syntactic_Features_Notebook_for_PAN_at_CLEF_2019 (дата обращения: 29.12.23).

20. Authorship Attribution of Social Media and Literary Russian-Language Texts Using Machine Learning Methods and Feature Selection / A. Fedotova, A. Romanov, A. Kurtukova, A. Shelupanov // Future Internet. 2022. Vol. 14(1). Iss. 4. 24 p.

21. Authorship identification of a russian-language text using support vector machine and deep neural networks / A. Romanov, A. Kurtukova, A. Fedotova [et al.] // Future Internet. 2021. Vol. 13(3). Iss. 1. 16 p.

22. Authorship Identification of Binary and Disassembled Codes Using NLP Methods / A. Romanov, A. Kurtukova, A. Fedotova, A. Shelupanov // Information. 2023. Vol. 14. Iss. 7. 361. 21 p.

23. Automated language-independent authorship verification (for Indo-European languages) / S. Adamovic, V. Miskovic, M. Milosavljevic [et al.] // Journal of the Association for Information Science and Technology. 2019. Vol. 70, no. 8. P. 858–871.

24. Babakov N. Beyond plain toxic: detection of inappropriate statements on flammable topics for the Russian language. URL: https://arxiv.org/pdf/2203.02392.pdf (дата обращения: 29.12.23).

25. Babakov N. Detecting inappropriate messages on sensitive topics that could harm a company's reputation / N. Babakov, V. Logacheva, O. Kozlova [et al.] // Proceedings of the 8th BSNLP Workshop on Balto-Slavic Natural Language Processing, BSNLP 2021-Co-located with the 16th European Chapter of the Association for Computational Linguistics, EACL 2021. 2021. P. 26–36.

26. Balakrishnan R. Protecting Anonymous speech: a generative adversarial network methodology for Removing Stylistic Indicators in text. URL: https://arxiv.org/pdf/2110.09495.pdf (дата обращения: 31.12.23).

27. BARD. URL: https://bard.google.com/?hl=ru (дата обращения: 29.12.23).

28. Barlas G., Stamatatos E. Cross-Domain Authorship Attribution Using Pre-trained Language Models // IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham, 2020. P. 255–266.

29. Bartelds M. Improving Cross-domain Authorship Attribution by Combining Lexical and Syntactic Features. URL: https://core.ac.uk/download/pdf/227369748.pdf (дата обращения: 29.12.23).

30. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. URL: https://arxiv.org/abs/1810.04805(дата обращения: 29.12.23).

31. Gender Identification in Russian Texts / R. Bhargava, G. Goel, A. Shah, Y. Sharma // FIRE (Working Notes). 2017. P. 13–16.

32. Deep bayes factor scoring for authorship verification / B. Boenninghoff, J. Rupp, R.M. Nickel, D. Kolossa // arXiv preprint arXiv:2008.10105. 2020. URL: https://arxiv.org/abs/2008.10105 (дата обращения: 29.12.23).

33. Boenninghoff B. Explainable Authorship Verification in Social Media via Attention-based Similarity Learning / B. Boenninghoff, S. Hessler, D. Kolossa, R.M. Nickel // 2019 IEEE International Conference on Big Data (Big Data). IEEE, 2019. P. 36–45.

34. Bogdanova A. Source code authorship attribution using file embeddings // Companion Proceedings of the 2021 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity. 2021. P. 31–33.

35. Authorship attribution of source code: a language-agnostic approach and applicability in software engineering / E. Bogomolov, V. Kovalenko, Y. Rebryk [et al.] // In Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2021. P. 932–944.

36. ICodeNet – A Hierarchical Neural Network Approach For Source Code Author Identification / P. Bora, T. Awalgaonkar, H. Palve [et al.] // In Proceedings of the 2021 13th International Conference on Machine Learning and Computing (ICMLC '21). Association for Computing Machinery, New York, NY, USA. 2021. P. 180–185.

37. Borboudakis G., Tsamardinos I. Forward-backward selection with early dropping // The Journal of Machine Learning Research. 2019. Vol. 20, no. 1. P. 276–314.

38. Boumber D. Experiments with convolutional neural networks for multi-label authorship attribution. URL: https://aclanthology.org/L18-1409.pdf (дата обращения: 29.12.23).

39. Breiman L. Random Forests // Machine Learning. 2001. Vol. 45. P. 5–32.

40. Britt C.L., Rocque M., Zimmerman G.M. The analysis of bounded count data in criminology // Journal of Quantitative Criminology. 2018. Vol. 34, no. 2. P. 591–607.

41. Buintsev D.N., Shelupanov A.A., Shevtsova O.O. Analysis of the use of obfuscating transformations for software // Inform. Secur. Is. 2015. Vol. 3. P. 38–43.

42. Burrows S., Uitdenbogerd A., Turpin A. Application of information retrieval techniques for source code authorship attribution // In Proceedings of the 14th International Conference on Database Systems for Advanced Applications. 2009. P. 699–713.

43. Caliskan-Islam A., Dauber E., Harang R. Git blame who? URL: https://arxiv.org/pdf/1701.05681 (дата обращения: 30.12.2023).

44. Caliskan-Islam A., Harang R., Liu A. Deanonymizing programmers via code stylometry // In Proceedings of the 24th USENIX Security Symposium. 2015. P. 255–270.

45. Chaka C. Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools URL: https://journals.sfu.ca/jalt/index.php/jalt/article/view/861 (дата обращения: 29.12.23).

46. ChatGPT. URL: https://chat.openai.com/ (дата обращения: 29.12.23).

47. ChatGPT4-o. URL: https://openai.com/index/hello-gpt-4o/ (дата обращения 19.05.2024).

48. Chechneva N. Parametric Adjectives in the Context of Sentiment Analysis // CMCL. 2020. P. 43–56.

49. Chen S.F.J., Goodman J. An empirical study of smoothing techniques for language modeling // Computer Speech & Language. 1999. Vol. 13, no. 4. P. 359–393.

50. Chowdhury H.A., Imon M.A.H., Islam M.S. Authorship attribution in Bengali Literature Using fastText's Hierarchical Classifier // 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT). IEEE, 2018. P. 102–106.

51. Clinski-Harabasz score. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz _score.html (дата обращения: 19.10.2023).

52. CodeBERT. URL: https://github.com/microsoft/CodeBERT (дата обращения: 30.05.2024).

53. Complex Cases of Source Code Authorship Identification Using a Hybrid Deep Neural Network / A. Kurtukova, A. Romanov, A. Shelupanov, A. Fedotova // Future Internet. 2022. Vol. 14. Iss. 287. 20 p.

54. Corbara S., Moreo A. Forging the Forger: An Attempt to Improve Authorship Verification via Data Augmentation // arXiv preprint. 2024. URL: https://arxiv.org/pdf/2403.11265 (дата обращения: 22.12.2024).

55. Cox D.R., Snell E.J. Analysis of Binary Data // 2nd Edition, Chapman and Hall/CRC, London. 1989. 248 p.

56. Cpp Guard. URL: https://github.com/techtocore/Cpp-Guard (дата обращения: 20.02.2024).

57. Damerau F.J. A technique for computer detection and correction of spelling errors // Commun. ACM. 1964. Vol. 7, no. 3. P. 171–176.

58. Daneshvar S. User Modeling in Social Media: Gender and Age Detection. University of Ottawa. 2019. URL: https://ruor.uottawa.ca/handle/10393/39535.

59. Dark Web Forums. URL: https://www.azsecure-data.org/dark-web-forums.html.

60. Davies-Bouldin score. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score. html (дата обращения: 19.10.2023).

61. Deep_translator’s documentation. URL: https://deep-translator.readthedocs.io/ (дата обращения: 19.10.2023).

62. Dementieva D., Babakov N., Panchenko A. MultiParaDetox: Extending Text Detoxification with Parallel Data to New Languages. URL: https://arxiv.org/pdf/2404.02037v1 (дата обращения: 24.05.2024).

63. Demsar J. Statistical Comparisons of Classifiers over Multiple Data Sets // Journal of Machine Learning Research. 2006. Vol. 7. P. 1–30.

64. Determining the Age of the Author of the Text Based on Deep Neural Network Models / A.S. Romanov, A.V. Kurtukova, A.A. Sobolev [et al.] // Information. 2020. Vol. 11. Iss. 12. 589. 12 p.

65. Deutsch C., Paraboni I. Authorship attribution using author profiling classifiers // Natural Language Engineering. 2023. Vol. 29, no. 1. P. 110–137.

66. Digital Authorship Attribution in Russian-Language Fanfiction and Classical Literature / A. Fedotova, A. Romanov, A. Kurtukova, A. Shelupanov // Algorithms. 2023. Vol. 16. Iss. 1. 13. 32 p.

67. Distributed Representations of Words and Phrases and their Compositionality / T. Mikolov, I. Sutskever, K. Chen [et al.] // Advances in Neural Information Processing Systems. 2013. Vol. 26.

68. Dmitrin Y.V. Comparison of deep neural network architectures for authorship attribution of Russian social media texts. URL: https://elibrary.ru/item.asp?id=35719542 (дата обращения: 20.02.24).

69. Du Z. All nlp tasks are generation tasks: A general pretraining framework. URL: https://arxiv.org/abs/2103.10360 (дата обращения: 20.02.2024).

70. Embarcadero-Ruiz D. Graph-Based Siamese Network for Authorship Verification // Mathematics. 2022. Vol. 10, no. 2. P. 1–24.

71. Embarcadero-Ruiz D., Gómez-Adorno H., Reyes-Hernández I. Graph-Based Siamese Network for Authorship Verification // Mathematics. 2022. Vol. 10, no. 2. P. 1–24.

72. Emotion detection. URL: https://huggingface.co/cointegrated/rubert-tiny2-cedr-emotion-detection (дата обращения 30.10.2023).

73. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling / J. Chung, C. Gulcehre, K. Cho, Y. Bengio // arXiv preprint arXiv: 1412.3555. 2014. URL: https://arxiv.org/abs/1412.3555 (дата обращения: 19.10.2023).

74. Enriching Word Vectors with Subword Information / P. Bojanowski, E. Grave, A. Joulin, T. Mikolov // Transactions of the Association for Computational Linguistics. 2017. Vol. 5. P. 135–146.

75. Evert S. Understanding and explaining Delta measures for authorship attribution // Digital Scholarship in the Humanities. 2017. Vol. 32, no. 2. P. 4–16.

76. Farnadi G. User Profiling Using Hinge-loss Markov Random Fields. URL: https://arxiv.org/abs/2001.01177 (дата обращения: 30.10.2023).

77. Feature Selection for High Dimensional Data Using Supervised Machine Learning Techniques / K. Lazaros, S. Tasoulis. A. Vrahatis, V. Plagianakos // 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan. 2022. P. 3891–3894.

78. Feature selection using genetic algorithm to improve classification in network intrusion detection system / A. Ferriyan, H. Thamrin, K. Takeda, J. Murai // 2017 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), Surabaya, Indonesia. 2017. P. 46–49.

79. Fisher R.A. The arrangement of field experiments // Journal of the Ministry of Agriculture. 1926. Vol. 33. P. 503–515.

80. Fix E., Hodges J.L. Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties // Technical Report: USAF School of Aviation Medicine, Randolph Field, Texas. 1951.

81. Floridi L., Chiriatti M. GPT-3: Its nature, scope, limits, and consequences // Minds and Machines. 2020. Vol. 3, no. 4. P. 681–694.

82. Franco-Salvador M. Subword-based Deep Averaging Networks for Author Profiling in Social Media. URL: https://pan.webis.de/downloads/publications/papers/francosalvador_2017.pdf (дата обращения: 08.03.2024).

83. Frantzeskou G., Stamatatos E., Gritzalis S. Identifying authorship by byte-level n-grams: The source code author profile (SCAP) method // Int. J. Digit. Evid. 2007. Vol. 1. P. 1–18.

84. Garain A. GRA_Net: A deep learning model for classification of age and gender from facial images // IEEE Access. 2021. Vol. 9. P. 85672–85689.

85. Gender, genre, and writing style in formal written texts / S. Argamon, M. Koppel, J. Fine, A.R. Shimoni // Text & talk. 2003. Vol. 23, no. 3. P. 321–346.

86. Ghiassi M. Automated text classification using a dynamic artificial neural network model // Expert Systems with Applications. 2012. Vol. 3, no. 12. P. 10967–10976.

87. GigaChat. URL: https://developers.sber.ru/portal/products/gigachat (дата обращения 08.03.2024).

88. GitHub Copilot. URL: https://copilot.GitHub.com (дата обращения 20.02.2024).

89. Gitlab. URL: https://about.gitlab.com/ (дата обращения: 20.02.2024).

90. Gorman R. Author identification of short texts using dependency treebanks without vocabulary // Digital Scholarship in the Humanities. 2020. Vol. 35, no. 4. P. 812–825.

91. FBSEM: a novel feature-based stacked ensemble method for sentiment analysis / Y. Görmez [et al.]. URL: http://www.mecs-press.org/ijitcs/ijitcs-v12-n6/IJITCS-V12-N6-2.pdf (дата обращения: 30.12.24).

92. Using Machine Learning Methods to Establish Program Authorship / S. Gorshkov, S. Nered, E. Ilyushin, D. Namiot // Int. J. Open Inf. Technol. 2019. Vol. 7. P. 2307–8162.

93. Graves A., Schmidhuber C. Framewise phoneme classification with bidirectional LSTM and other neural network architectures // Neural Networks. 2005. Vol. 18, no. 5. P. 602–610.

94. Grieve J. Quantitative authorship attribution: An evaluation of techniques // Literary and linguistic computing. 2007. Vol. 22, no. 3. P. 251–270.

95. Gröndahl T. Effective writing style imitation via combinatorial paraphrasing. URL: https://arxiv.org/abs/1905.13464 (дата обращения: 29.12.23).

96. Halvani O. Cross-domain authorship verification based on topic agnostic features. 2020. URL: http://ceur-ws.org/Vol-2696/paper_114.pdf.

97. Hashemi A. Enhancing writing style change detection using transformer-based models and data augmentation. URL: https://downloads.webis.de/pan/publications/papers/hashemi_2023.pdf (дата обращения: 29.12.23).

98. HDBSCAN. URL: https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html (дата обращения: 19.10.2023).

99. Hedegaard S., Simonsen J.G. Lost in translation: Authorship attribution using frame semantics // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011. P. 65–70.

100. Semantic-based features for author profiling identification: First insights / D. Hernández, R. Guzmán-Cabrera, A. Reyes, M.A. Rocha. URL: https://citeseerx.ist.psu.edu/viewdoc/ download?doi=10.1.1.667.8176&rep=rep1&type=pd (дата обращения: 19.10.2023).

101. Hinh R., Shin S., Taylor J. Using frame semantics in authorship attribution // 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2016. P. 004093–004098.

102. Hochreiter S., Schmidhuber J. Long Short-term Memory // Neural computation. 1997. Vol. 9. P. 1735–1980.

103.How ISIS Uses Twitter. URL: https://www.kaggle.com/datasets/fifthtribe/how-isis-uses-twitter (дата обращения: 19.10.2023).

104. TDRLM: Stylometric learning for authorship verification by Topic-Debiasing / X. Hu, W. Ou, S. Acharya [et al.] // Expert Syst. Appl. 2023. Vol. 233. 120745 p.

105. Huang W., Su R., Iwaihara M. Contribution of improved character embedding and latent posting styles to authorship attribution of short texts // Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data. Springer, Cham, 2020. P. 261–269.

106. Huang B., Chen C., Shu K. Can Large Language Models Identify Authorship? 2024. URL: https://arxiv.org/abs/2403.08213 (дата обращения: 30.12.2024).

107. Hugging Face. A Family of Pretrained Transformer Language Models for Russian. URL: https://huggingface.co/ai-forever/rugpt3large_based_on_gpt2 (дата обращения: 29.12.23).

108. Iliev R., Dehghani M., Sagi E. Automated text analysis in psychology: Methods, applications, and future developments // Language and cognition. 2015. Vol. 7, no. 2. P. 265–290.

109. Ishihara S., Carne M. Likelihood ratio estimation for authorship text evidence: An empirical comparison of score-and feature-based methods. URL: https://www.sciencedirect.com/science/ article/pii/S0379073822000986 (дата обращения 05.04.2023).

110. Analysis of textual content as a mechanism for ensuring safety of the socio-cyberphysical system / A. Iskhakova, R. Meshcheryakov, A. Iskhakov, I. Kulagina // 2021 International Siberian Conference on Control and Communications (SIBCON). IEEE, 2021. P. 1–6.

111. Jafariakinabad F., Hua K.A. A Self-supervised Representation Learning of Sentence Structure for Authorship Attribution // ACM Transactions on Knowledge Discovery from Data (TKDD). 2022. Vol. 16, no. 4. P. 1–16.

112. Jasper. URL: https://www.jasper.ai/ (дата обращения: 29.12.23).

113. Johansson F., Isbister T. FOI Cross-Domain Authorship Attribution for Criminal Investigations. 2019. URL: https://ceur-ws.org/Vol-2380/paper_198.pdf (дата обращения: 08.03.2024).

114. Joshi B., Khani S.H., HabibiLashkari A. AuthAttLyzer-V2: Unveiling Code Authorship Attribution using Enhanced Ensemble Learning Models & Generating Benchmark Dataset. URL: https://arxiv.org/abs/2406.19896 (дата обращения: 30.12.2024).

115. JS Obfuscator Tool. URL: https://obfus-cator.io/ (дата обращения: 20.02.2024).

116. JS-Obfuscator. URL: https://github.com/caiguanhao/js-obfuscator (дата обращения: 20.02.2024).

117. Kacmarcik G., Gamon M. Obfuscating document stylometry to preserve author anonymity // Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions. 2006. P. 444–451.

118. Kapitanov A.I. Approach to automatic identification of terrorist and radical content in social networks messages // 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). IEEE, 2018. P. 1517–1520.

119. Karami A. Automatic Categorization of LGBT User Profiles on Twitter with Machine Learning. URL: https://www.mdpi.com/2079-9292/10/15/1822 (дата обращения: 29.12.23).

120. Katz S.M. Estimation of probabilities from sparse data for the language model component of a speech recognizer // IEEE Transactions on Acoustics, Speech and Signal Processing. 1987. Vol. 35, no. 3. P. 400–401.

121. Khatun A. Authorship Attribution in Bangla literature using Character-level CNN // 2019 22nd International conference on computer and information technology (ICCIT). IEEE, 2019. P. 1–5.

122. Kovalev A.K., Kuznetsova Yu.M. Possibilities of automatic text analysis in the task of determining the psychological characteristics of the author // Experimental Psychology. 2020. Vol. 13, no. 1. P. 149–158.

123. Kumar B.A., Misra N.K. Masked face age and gender identification using CAFFE-modified MobileNetV2 on photo and real-time video images by transfer learning and deep learning techniques. URL: https://www.sciencedirect.com/science/article/pii/S0957417424000447 (дата обращения: 25.05.2024).

124. Kurtukova A., Romanov A., Shelupanov A. Source Code Authorship Identification Using Deep Neural Networks // Symmetry. 2020. Vol. 12. Iss. 12. 2044. 17 p.

125. Kurtukova A., Romanov A., Fedotova A. De-Anonymization of the Author of the Source Code Using Machine Learning Algorithms // Conference Proceedings of 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON). 2019. P. 0612–0617.

126. Labbé C. Ike Antkare one of the great stars in the scientific firmament. URL: https://hal.science/file/index/docid/713564/filename/TechReportV2.pdf (дата обращения: 02.01.2024).

127. LaBSE. URL: https://huggingface.co/sentence-transformers/LaBSE (дата обращения: 19.10.2023).

128. Language and Obfuscation Oblivious Source Code Authorship Attribution / Z. Zafar, M.U. Sarwar, S. Salem, M.Z. Malik // IEEE Access. 2020. Vol. 8. P. 197581–197596.

129. Lanza-Cruz I., Berlanga R., Aramburu M.J. Multidimensional author profiling for social business intelligence // Information Systems Frontiers. 2024. Vol. 26, no. 1. P. 195–215.

130. LeCun Y., Bengio Y., Hinton G. Deep learning // Nature. 2015. Vol. 521. P. 436–444.

131. Lee J.S., Hsiang J. Patent claim generation by fine-tuning OpenAI GPT-2 // World Patent Information. 2020. Vol. 62. 101983 p.

132. Le-Khac P.H., Healy G., Smeaton A.F. Contrastive Representation Learning: A Framework and Review // arXiv preprint: 2010.05113. 2020. URL: https://arxiv.org/abs/2010.05113 (дата обращения: 20.02.2024).

133. Li X., Prasad C. Effectively teaching coding standards in programming // In Proceedings of the 6th Conference on Information Technology Education-SIGITE. 2005. P. 239–244.

134. Linux kernel. URL: https://github.com/torvalds/linux (дата обращения: 20.02.2024).

135. Litvinova T., Litvinova O., Panicheva P. Authorship attribution of Russian forum posts with different types of n-gram features // Proceedings of the 2019 3rd International Conference on Natural Language Processing and Information Retrieval. 2019. P. 9–14.

136. Lloyd S. Least squares quantization in PCM // IEEE Transactions on Information Theory. 1982. Vol. 28, no. 2. P. 129–137.

137. Luhn H.P. A Statistical Approach to Mechanized Encoding and Searching of Literary Information // IBM Journal of Research and Development. 1957. Vol. 1, no. 4. P. 309–317.

138. Lundberg S., Su-In L. A Unified Approach to Interpreting Model Predictions // NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017. P. 4768–4777.

139. Mahbub P., Oishie N.Z., Haque R. Authorship Identification of Source Code Segments Written by Multiple Authors Using Stacking Ensemble Method // Proceedings

of the 2019 22nd International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh. 2019. P. 1–6.

140. A Girl Has No Name: Automated Authorship Obfuscation using Mutant-X / A. Mahmood, F. Ahmad, Z. Shafiq [et al.] // Proc. Priv. Enhancing Technol. 2019. Vol. 2019, no. 4. P. 54–71.

141. Makarova M. Authorship attribution for Differences between Literary Texts by Bilingual Russian-French and Non-Bilingual French Authors. URL: https://arxiv.org/abs/2303.13622 (дата обращения: 29.04.2024).

142. Mandalam A.V. Embedding-based Authorship Identification of Source Code // Forum for Information Retrieval Evaluation, December 16–20, 2020, Hyderabad, India. P. 31–33.

143. Manikandan D., Subramanian M., Shanmugavadivel K. A System For Detecting Abusive Contents Against LGBT Community Using Deep Learning Based Transformer Models. URL: https://ceur-ws.org/Vol-3395/T2-4.pdf (дата обращения: 10.03.2023).

144. Markov I., Gómez-Adorno H., Sidorov G. The winning approach to cross-genre gender identification in russian at rusprofiling 2017 // Training. 2017. Vol. 600. P. 1–216.

145. McCulloch W.S., Pitts W. A logical calculus of the ideas immanent in nervous activity // Bulletin of Mathematical Biophysics. 1943. Vol. 5. P. 115–133.

146. Mclachlan G. Mahalanobis Distance // Resonance. 1990. Vol. 4. P. 20–26.

147. Author Profiling: Age Prediction Based on Advanced Bayesian Networks / S. Metchi, M. Jaoua, R. Faiz [et al.] // Res. Comput. Sci. 2016. Vol. 110. P. 129–137.

148. Meilă M., Shi J. Learning Segmentation by Random Walks // Neural Information Processing Systems. 2001. Vol. 13.

149. Moshkin V., Fadeev D., Yarushkina N. Development of a System for Finding Extremist Texts // CEUR Workshop Proceedings. 2021. P. 45–52.

150. MuCAAT: Multilingual Contextualized Authorship Anonymization of Texts from social networks / V. Panov, M. Kovalchuk, A. Filatova, S. Teryoshkin // Procedia Computer Science. 2022. Vol. 212. P. 322–329.

151. Mussiraliyeva S. On detecting online radicalization and extremism using natural language processing // 2020 21st International Arab Conference on Information Technology (ACIT). IEEE, 2020. P. 1–5.

152. Najafi M., Tavan E. Text-to-Text Transformer in Authorship Verification Via Stylistic and Semantical Analysis. URL: https://ceur-ws.org/Vol-3180/paper-215.pdf (дата обращения: 10.03.2023).

153. Navada A. Overview of use of decision tree algorithms in machine learning // IEEE Control and System Graduate Research Colloquium (ICSGRC). Shah Alam, Malaysia. 2011. P. 37–42.

154. Neculoiu P., Versteegh M., Rotaru M. Learning Text Similarity with Siamese Recurrent Networks // Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, Germany, August 11th, 2016. P. 148–157.

155. Neural Network Model to Include Textual Dependency Tree Structure in Gender Classification of Russian Text Author / A. Sboev, A. Selivanov, R. Rybka [et al.] // Advanced Technologies in Robotics and Intelligent Systems. Springer, Cham, 2020. P. 405–412.

156. Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment / D. Nguyen, D. Trieschnigg, A.S. Doğruöz [et al.] // 25th International Conference on Computational Linguistics (COLING 2014). Dublin City University and Association for Computational Linguistics, 2014. P. 1950–1961.

157. Nishiyama K., Adebayo G.O., Yampolskiy R. Authorship Identification of Translational Algorithms // 2021 IEEE 15th International Conference on Semantic Computing (ICSC). IEEE, 2021. P. 90–91.

158. Nitu M., Dascalu M. Authorship Attribution in Less-Resourced Languages: A Hybrid Transformer Approach for Romanian // Applied Sciences. 2024. Vol. 14. Iss. 7. 2700. URL: https://www.mdpi.com/2076-3417/14/7/2700 (дата обращения: 30.12.2024).

159. NLTK. URL: https://www.nltk.org/ (дата обращения: 18.01.24).

160. Application of neural network modeling in the task of destructive content detecting / V. Okhapkin, E. Okhapkina, A. Iskhakova, A. Iskhakov // CEUR Workshop

Proc. Proc. 3rd Int. Conf. R. Piotrowski’s Read. Lang. Eng. Appl. Linguist., PRLEAL. 2019. P. 85–94.

161. Open AI. URL: https://openai.com/blog/openai-codex (дата обращения: 20.02.2024).

162. Opy. URL: https://github.com/QQuick/Opy (дата обращения: 20.02.2024).

163. Deep neural network-based classification model for Sentiment Analysis / D. Pan, J. Yuan, L. Li, D. Sheng // 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC). IEEE, 2019. P. 1–4.

164. Panicheva P., Litvinova T. Authorship attribution in Russian in real-world forensics scenario // International Conference on Statistical Language and Speech Processing. Cham: Springer International Publishing, 2019. P. 299–310.

165. Panicheva P., Mirzagitova A., Ledovaya Y. Semantic feature aggregation for gender identification in Russian Facebook // Conference on Artificial Intelligence and Natural Language. Springer, Cham, 2017. P. 3–15.

166. Multilingual contextual affective analysis of LGBT people portrayals in Wikipedia / C.Y. Park, X. Yan, A. Field, Y. Tsvetkov // Proceedings of the International AAAI Conference on Web and Social Media. 2021. Vol. 15. P. 479–490.

167. Patchala J., Bhatnagar R. Authorship attribution by consensus among multiple features // Proceedings of the 27th International Conference on Computational Linguistics. 2018. P. 2766–2777.

168. Pearson K. On lines and planes of closest fit to systems of points in space // The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901. Vol. 2, no. 11. P. 559–572.

169. Pennington J., Socher R., Manning C. GloVe: Global Vectors for Word Representation // In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha. 2014. P. 1532–1543.

170. PHP Obfuscator. URL: https://github.com/naneau/php-obfuscator (дата обращения: 20.02.2024).

171. Pimonova E., Durandin O., Malafeev A. Authorship Attribution in Russian with New High-Performing and Fully Interpretable Morpho-Syntactic Features //

International Conference on Analysis of Images, Social Networks and Texts. Cham: Springer International Publishing, 2019. P. 193–204.

172. Pisarevskaya D., Shavrina T. WikiOmnia: generative QA corpus on the whole Russian Wikipedia. URL: //https://arxiv.org/abs/2204.08009 (дата обращения: 20.02.2024).

173. Pizarro J. Using N-grams to detect Bots on Twitter. URL: https://ceur-ws.org/Vol-2380/paper_183.pdf (дата обращения: 28.04.2024).

174. PolyCoder. URL: https://venturebeat.com/2022/03/04/researchers-open-source-code-generating-ai-they-claim-can-beat-openais-codex/ (дата обращения: 20.02.2024).

175. Popa M. Techniques of Program Code Obfuscation for Secure Software // J. Mob. Embed. Distrib. Syst. 2011. Vol. 3. P. 205–219.

176. Potha N., Stamatatos E. Improved algorithms for extrinsic author verification // Knowledge and Information Systems. 2020. Vol. 62, no. 5. P. 1903–1921.

177. Potthast M., Hagen M., Stein B. Author Obfuscation: Attacking the State of the Art in Authorship Verification // CLEF (Working Notes). 2016. P. 716–749.

178. Pyarmor. URL: https://github.com/dashingsoft/pyarmor (дата обращения: 20.02.2024).

179. Quiring E., Maier A., Rieck K. Misleading authorship attribution of source code using adversarial learning // 28th USENIX Security Symposium (USENIX Security 19). 2019. P. 479–496.

180. Radford A., Karthik N. Improving Language Understanding by Generative Pre-Training (2018). URL: https://ulaval-damas.github.io/glo4030/assets/slides/3.2-GPT.pdf (дата обращения: 18.01.24).

181. Ranaldi L. Shedding Light on the Dark Web: Authorship Attribution in Radical Forums. URL: https://www.mdpi.com/2078-2489/13/9/435.

182. Rangel F., Rosso P. On the impact of emotions on author profiling // Information processing & management. 2016. Vol. 52, no. 1. P. 73–92.

183. Religious Texts Used By ISIS. URL: https://www.kaggle.com/datasets/fifthtribe/isis-religious-texts (дата обращения: 19.10.2023).

184. Rocha A. Authorship attribution for social media forensics // IEEE transactions on information forensics and security. 2016. Vol. 12, no. 1. P. 5–33.

185. Linguistic Modeling as a Basis for Creating Authorship Attribution Software / A. Romanov, A. Khomenko, Y. Baranova, K. Zadvornov // Computational Linguistics and Intellectual Technologies. 2021. Vol. 20. P. 18–27.

186. Natural text anonymization using universal transformer with a self-attention / A. Romanov, A. Kurtukova, A. Fedotova, R. Meshcheryakov // Proceedings of the III International Conference on Language Engineering and Applied Linguistics (PRLEAL-2019), Saint Petersburg, Russia. 2019. P. 22–37.

187. Age determination of the social media post's author using deep neural networks and facial processing models / A. Romanov, A. Kurtukova, A. Sobolev, A. Fedotova // CEUR Workshop Proceedings. 2021. Vol. 2842. P. 51–59.

188. Rumelhart D.E., Hinton G.E., Williams R.J. Learning internal representations by error propagation // Tech. rep. ICS 8504. San Diego, California: Institute for Cognitive Science, University of California. 1985. 43 p.

189. Russian sensitive topics. URL: https://huggingface.co/aanc/russian-sensitive-topics (дата обращения: 30.10.2023).

190. Russian toxicity classifier. URL: https://huggingface.co/s-nlp/russian_toxicity_classifier (дата обращения: 30.10.2023).

191. Russian Twitter Corpus. URL: https://www.kaggle.com/datasets/maximsuvorov/rutweetcorp (дата обращения: 20.02.2024).

192. Sber AI ruGPT-3. URL: https://developers.sber.ru/portal/tools/rugpt-3 (дата обращения: 20.02.2024).

193. Semantic Clustering and Transfer Learning in Social Media Texts Authorship Attribution / A. Fedotova, A. Kurtukova, A. Romanov, A. Shelupanov // IEEE Access. 2024. Vol. 12. P. 39783–39803.

194. Sentence Transformers. URL: https://huggingface.co/sentence-transformers (дата обращения: 19.10.2023).

195. Shijaku R., Canhasi E. ChatGPT Generated Text Detection. 2023. URL: https://www.researchgate.net/publication/366898047_ChatGPT_Generated_Text_Detection (дата обращения: 20.02.24).

196. Shmelova T. Machine learning and text analysis in an artificial intelligent system for the training of air traffic controllers // Research Anthology on Reliability and Safety in Aviation Systems, Spacecraft, and Air Transport. IGI Global, 2021. P. 237–286.

197. Sierra S. Automatic authorship analysis using deep neural networks // Ingeniería de Sistemas. 2018. URL: https://repositorio.unal.edu.co/handle/unal/76747 (дата обращения: 29.12.23).

198. Silhouette score. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html (дата обращения: 19.10.2023).

199. Simko L., Zettlemoyer L., Kohno T. Recognizing and Imitating Programmer Style: Adversaries in Program Authorship Attribution // Proceedings on Privacy Enhancing Technologies. 2018. 10.1515/popets-2018-0007.

200. Skrylnikov S.S., Posokhov P.A., Makhnytkina O. Artificial text detection in Russian language: a BERT-based Approach // Dialogue-2022. URL: https://doi.org/10.28995/2075-7182-2022-21-470-476.

201. Soesanto A.M., Chandra V.C., Suhartono D. Sentiments comparison on twitter about lgbt // Procedia Computer Science. 2023. Vol. 216. P. 765–773.

202. Spacy. URL: https://spacy.io/ (дата обращения: 18.01.24).

203. Sparck Jones K. A statistical interpretation of term specificity and its application in retrieval // Journal of documentation. 1972. Vol. 28, no. 1. P. 11–21.

204. Stanford NLP. URL: https://nlp.stanford.edu/ (дата обращения: 18.01.24).

205. Uchendu A. Authorship attribution for neural text generation // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020. P. 8384–8395.

206. Ullah F. Source code authorship attribution using hybrid approach of program dependence graph and deep learning model // IEEE Access. 2019. Vol. 7. P. 141987–141999.

207. UMAP. URL: https://umap-learn.readthedocs.io/en/latest/ (дата обращения: 19.10.2023).

208. Van Tussenbroek T. Who said that? Comparing performance of TF-IDF and fastText to identify authorship of short sentences. URL: https://repository.tudelft.nl/islandora/object/uuid:93873bbf-2886-4023-b696-e11be2b99024 (дата обращения: 19.10.2023).

209. Vanmassenhove E. Getting gender right in neural machine translation. URL: https://arxiv.org/abs/1909.05088 (дата обращения: 19.10.2023).

210. Vapnik V. Statistical Learning Theory // New York: Wiley. 1998. 732 p.

211. Varela P., Justino E., Oliveira L.S. Selecting syntactic attributes for authorship attribution // The 2011 International Joint Conference on Neural Networks. IEEE, 2011. P. 167–172.

212. Vashisth P., Meehan K. Gender classification using twitter text data // 31st Irish Signals and Systems Conference (ISSC). IEEE, 2020. P. 1–6.

213. Wang N., Ji S. Integration of Static and Dynamic Code Stylometry Analysis for Programmer De-anonymization // In Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security. 2018. P. 74–84.

214. Wang Y., Zheng B., Huang H. Complying with Coding Standards or Retaining Programming Style: A Quality Outlook at Source Code Level // JSEA. 2008. Vol. 1. P. 88–91.

215. Weerasinghe J. Feature Vector Difference based Neural Network and Logistic Regression Models for Authorship Verification. 2020. URL: https://par.nsf.gov/biblio/10249189 (дата обращения: 19.10.2023).

216. Wisse W., Veenman C.J. Scripting DNA: Identifying the JavaScript Programmer // Digit. Investig. 2015. Vol. 15. P. 61–71.

217. A prototype application to identify LGBT patients in clinical notes / T.E. Workman, J.L. Goulet, C. Brandt [et al.] // 2020 IEEE International Conference on Big Data (Big Data). IEEE, 2020. P. 4270–4275.

218. Wu H., Zhang Z., Wu Q. Exploring syntactic and semantic features for authorship attribution // Applied Soft Computing. 2021. Vol. 111. P. 107815–107822.

219. Yakpro-po. URL: https://github.com/pkfr/yakpro-po (дата обращения: 20.02.2024).

220. YandexGPT 2. URL: https://ya.ru/gpt/2 (дата обращения: 29.12.23).

221. YandexGPT 3. URL: https://ya.ru/ai/gpt-3 (дата обращения: 19.05.24).

222. Yang M. A topic drift model for authorship attribution // Neurocomputing. 2018. Vol. 273. P. 133–140.

223. Authorship attribution of source code by using backpropagation neural network based on particle swarm optimization / X. Yang, Q. Li, Y. Guo, M. Zhang. URL: https://journals.plos.org/plosone/-article?id=10.1371/journal.pone.0187204 (дата обращения: 30.12.23).

224. Yao G., Xiaojian H., Guanxiong W. A novel ensemble feature selection method by integrating multiple ranking information combined with an SVM ensemble model for enterprise credit risk prediction in the supply chain // Expert Systems with Applications. 2022. 117002 p.

225. Supervised Contrastive Learning for Multi-Author Writing Style Analysis / Z. Ye, C. Zhong, H. Qi, Y. Han. URL: https://downloads.webis.de/pan/publications/papers/ye_2023.pdf (дата обращения: 29.12.23).

226. Younkin V., Litvak M., Rabaev I. Automatic Gender Identification from Text // Applied Sciences. 2024. Vol. 14. Iss. 24. 12041 p. URL: https://www.mdpi.com/2076-3417/14/24/12041 (дата обращения: 30.12.2024).

227. KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations / F.M. Zanzotto, A. Santilli, L. Ranaldi [et al.] // Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). 2020. P. 256–267.

228. Zhen L. RoPGen: Towards Robust Code Authorship Attribution via Automatic Coding Style Transformation // In Proceedings of the 2022 IEEE 44th International Conference on Software Engineering (ICSE). Pittsburgh, PA, USA. 2022. P. 1906–1918.

229. Zia T., Ilyas M.I.J. Source Code Author Attribution Using Author's Programming Style and Code Smells // Intel. Syst. Appl. 2017. Vol. 5. P. 27–33.

230. Андреев И.А. Исследование методов и алгоритмов обработки текстовой информации социальных сетей в задачах формирования социального портрета пользователя: дис. канд. техн. наук: 05.13.01 / И.А. Андреев. Ульяновский государственный технический университет. Ульяновск, 2022. 166 с.

231. Антиплагиат. URL: https://antiplagiat.ru/ (дата обращения: 30.06.2024).

232. Архитектура интеллектуальной системы для идентификации автора исходного кода / А.В. Куртукова, А.С. Романов, А.А. Шелупанов, А.М. Федотова // Доклады ТУСУР. 2022. Т. 25, № 3. С. 39–44.

233. Атаева О.М., Серебряков В.А., Тучкова Н.П. Идентификация авторов в рамках предметной области в семантической библиотеке // Электронные библиотеки. 2021. Т. 24, № 2. С. 198–217.

234. Ахмедшин Р.Л., Енгалычев В.Ф. Исследование почерка и профилирование серийных преступлений: тактико-криминалистический аспект // Актуальные вопросы судебной психологической экспертизы и комплексной экспертизы с участием психолога. Перспективы научного и прикладного исследования. 2019. С. 62–67.

235. Батурин М.М., Белов Ю.С. Применение структурного анализа для определения авторства текста на русском языке без использования лексических признаков // Международный журнал прикладных и фундаментальных исследований. 2023. № 1. С. 30–36.

236. Библиотека Максима Мошкова. URL: http://lib.ru/ (дата обращения: 20.06.2022).

237. Буинцев Д.Н. Методы защиты программных средств на основе запутывающих преобразований: дис. … канд. техн. наук: 05.13.19 / Д.Н. Буинцев. Томский государственный университет систем управления и радиоэлектроники. 2006. 121 с.

238. Бутакова Л. Автороведческая экспертиза: трудности идентификации // Юрислингвистика. 2019. № 12. С. 22–28.

239. Вдовина Л.А. Атрибуция авторства на основе оценки стилеметрических параметров текста (на материале англоязычных туристических блогов): маг. дис. : 45.04.02 / Л.А. Вдовина. Сибирский федеральный университет. 2021. 123 с.

240. Воробьева А.А. Методика идентификации интернет-пользователя на основе стилистических и лингвистических характеристик коротких электронных сообщений: дис. … канд. техн. наук: 05.13.19 / А.А. Воробьева. Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики. СПб., 2017. 154 с.

241. Вычегжанин С.В. Автоматическое распознавание точки зрения автора текста на основе ансамблей методов машинного обучения: дис. … канд. техн. наук: 05.13.17 / С.В. Вычегжанин. Вятский государственный университет. 2021. 167 с.

242. Галкина Д.А. Автоматическое определение профиля автора текста на естественном языке // Путь в науку: прикладная математика, информатика и информационные технологии. 2021. С. 26–28.

243. Развитие методов и построение алгоритмов поиска и классификации деструктивного контента, циркулирующего в социальной сети / А.А. Гончаров, Е.Ю. Чапурин, В.И. Белоножкин [и др.] // Информация и безопасность. 2019. Т. 22, № 3. С. 345–360.

244. Гринин И.Л. Разработка, тестирование и сравнение моделей сентиментального анализа коротких текстов // Инновации и инвестиции. 2020. № 6. С. 186–189.

245. Давыдова Ю.В. Методы текстового поиска и обработки информации в социальных сетях при управлении деятельностью правоохранительных органов: дис. … канд. техн. наук, 05.13.10 / Ю.В. Давыдов. ФГБОУ ВО «Орловский государственный университет имени И. С. Тургенева». Белгород, 2021. 146 с.

246. Дубовик А.Р. Автоматическое определение стилистической принадлежности текстов по их статистическим параметрам // Компьютерная лингвистика и вычислительные онтологии. 2017. № 1. С. 29–45.

247. Единый реестр доменных имен, указателей страниц сайтов в информационно-телекоммуникационной сети «Интернет» и сетевых адресов,

позволяющих идентифицировать сайты в информационно-телекоммуникационной сети «Интернет», содержащие информацию, распространение которой в Российской Федерации запрещено. URL: https://eais.rkn.gov.ru/ (дата обращения: 21.01.2024).

248. Зиберова О.С. Профилирование террориста (экстремиста) с истероидным психотипом // Криминалистика – наука без границ: традиции и новации. 2021. С. 116–118.

249. Идентификация автора исходного кода программы на основе неоднородных данных для решения задач кибербезопасности / А.В. Куртукова, А.С. Романов, А.А. Шелупанов, А.М. Федотова // Моделирование, оптимизация и информационные технологии. 2022. № 10(3). URL: https://moitvivt.ru/ru/journal/pdf?id=1227.

250. Исаченко В.В., Апанович З.В. Система анализа и визуализации для кросс-языковой идентификации авторов научных публикаций // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 49–61.

251. Использование и развитие методик оценки тональности контентов социальных сетей в целях обеспечения информационной безопасности пользователей / Е.А. Москалева, Е.Р. Нежельский, М.Е. Волкова [и др.] // Информация и безопасность. 2019. Т. 22, № 3. С. 433–444.

252. Исхакова А.О. Метод и программное средство определения искусственно-созданных текстов: дис. … канд. техн. наук: 05.13.17 / А.О. Исхакова. ФББОУ ВО «Томский государственный университет систем управления и радиоэлектроники» (ТУСУР). 2016. 123 с.

253. Кислицына М.Ю., Орлов Ю.Н. Статистический анализ полного корпуса художественной литературы на русском языке и распознавание автора // Препринты Института прикладной математики им. М.В. Келдыша РАН. 2024. № 17. С. 17–24.

254. Коляда А.С., Гогунский В.Д. Достоверность идентификации авторства научных публикаций на основе латентно семантического анализа // Восточно-Европейский журнал передовых технологий. 2014. Т. 3, № 2(69). С. 36–40.

255. Комиссаров А.Ю. Криминалистическое исследование письменной речи с использованием ЭВМ: дис. … канд. юрид. наук : 12.00.09 / А.Ю. Комиссаров. М., 2001. 225 с.

256. Корней А.О. Методы и алгоритмы аспектного анализа тональности на основе гибридной семантико-статистической модели естественного языка: дис. канд. техн. наук: 05.13.17 / А.О. Корней. Федеральное государственное бюджетное учреждение науки Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук. Новосибирск, 2022. 134 с.

257. Кузнецова Е., Зиновьева Е. Психологические аспекты определения и изучения лидеров мнений в цифровой среде // Мир науки. Педагогика и психология 2020. № 5. С. 1–18. URL: https://mir-nauki.com/PDF/97PSMN520.pdf (дата обращения: 23.01.2024).

258. Куртукова А.В., Сваровская Е.Е., Романов А.С. Оценка влияния обфускации на процесс идентификации автора программного кода // Доклады ТУСУР. 2020. Т. 23, № 2. С. 50–54.

259. Куртукова А.В., Романов А.С. Идентификация автора исходного кода методами машинного обучения // Труды СПИИРАН. 2019. № 3(18). С. 741–765.

260. Куртукова А.В., Романов А.С. Моделирование архитектуры нейронной сети в задаче идентификации автора исходного кода // Доклады ТУСУР. 2019. Т. 22, № 3. С. 37–42.

261. Куртукова А.В., Романов А.С., Шелупанов А.А. Разработка методики идентификации авторства бинарных и дизассемблированных кодов программы на основе ансамбля современных методов обработки естественного языка // Доклады ТУСУР. Томск, 2023. Т. 26, № 4. С. 53–60.

262. Кутузов И.М. Алгоритмы и методы защиты программного кода на базе обфускации: дис. … канд. техн. наук: 05.13.19 / И.М. Кутузов. НИУ ИТМО, Санкт-Петербург. СПб., 2016. 117 с.

263. Литвинова Т.А. Профилирование автора текста как одно из стратегических направлений исследований // Современные проблемы гражданской защиты. 2013. № 1(6). С. 38–41.

264. Марусенко М.А. Атрибуция анонимных и псевдонимных литературных произведений методами теории распознавания образов. Л. : ЛГУ, 1990. 164 с.

265. Методика идентификации автора текстовых материалов деструктивной направленности / А.С. Романов, А.М. Федотова, А.В. Куртукова, А.А. Шелупанов // Информационные технологии. 2024. Т. 30, № 12. С. 632–640.

266. Методика определения возраста автора текста на основе метрик удобочитаемости и лексического разнообразия / А.А. Соболев, А.М. Федотова, А.В. Куртукова, А.С. Романов, А.А. Шелупанов // Доклады ТУСУР. Томск, 2022. Т. 25, № 2. С. 45–52.

267. Методы автоматического аннотирования и выделения ключевых слов в задачах обнаружения экстремисткой информации в сети Интернет / И.В. Машечкин, М.И. Петровский, И.И. Поспелова, Д.В. Царёв // Современные информационные технологии и ИТ-образование. 2016. Т. 12, № 1. С. 188–198.

268. Методы выявления по тексту психологических характеристик автора (на примере агрессивности) / А.К. Ковалёв, Ю.М. Кузнецова, А.Н. Минин [и др.] // Вопросы кибербезопасности. 2019. № 4(32). С. 72–79.

269. Методы машинного обучения для задачи обнаружения и мониторинга экстремистской информации в сети интернет / И.В. Машечкин, М.И. Петровский, Д.В. Царёв, М.Н. Чикунов // Программирование. 2019. № 3. С. 18–37.

270. Минаев В.А., Симонов А.В., Реброва А.Д. Автоматизированное выявление деструктивного контента в социальных медиа // Информационная безопасность: вчера, сегодня, завтра. 2021. С. 124–130.

271. Министерство юстиции Российской Федерации. Экстремистские материалы. URL: https://minjust.gov.ru/ru/extremist-materials, свободный (дата обращения: 25.01.2024).

272. Мониторинг и выявление деструктивных информационных воздействий в современных социальных медиа / В.А. Минаев, К.М. Бондарь, А.В. Симонов,

П.Б. Скрипко // Информационная безопасность: вчера, сегодня, завтра. 2022. С. 140–145.

273. Морозов Н.А. Лингвистические спектры: средство для отличения плагиатов от истинных произведений того или иного известного автора. Стилеметрический этюд // Известия отд. русского языка и словесности Имп. Акад. наук. 1915. Т. XX. Кн. 4. С. 93–127.

274. Москин Н.Д. Теоретико-графовые модели, методы и программные средства интеллектуального анализа текстовой информации на примере фольклорных и литературных произведений: дис. … д-ра техн. наук: 05.13.18 / Н.Д. Москин. Петрозаводский гос. университет. Петрозаводск, 2022. 346 с.

275. Никеров Д.М., Хохлова О.М. Преступления в сфере высоких технологий в современной России // Вестник Восточно-Сибирского института МВД России. 2019. № 2(89). С. 82–93.

276. Новый частотный словарь русской лексики. URL: http://dict.ruslang.ru/freq.php/ (дата обращения: 20.02.2024).

277. Огорелков И.В. Исследование лингвистических характеристик текста с целью определения пола автора на примере анализа письменных русскоязычных текстов политического дискурса: дис. канд. техн. наук: 10.02.01 / И.В. Огорелков. ФГБОУ ВО «Государственный институт русского языка им. А.С. Пушкина». М., 2021. 457 с.

278. Огорелков И.В. Исследование лингвистических характеристик письменного текста политического дискурса с целью определения пола автора // Язык. Право. Общество: сб. ст. V Междунар. науч.-практ. конф. (г. Пенза, 22–25 мая 2018 г.). Пенза : Изд-во ПГ, 2018. С. 88–93.

279. Онипенко Н.К. Теория коммуникативной грамматики и проблема системного описания русского синтаксиса // Русский язык в научном освещении. 2001. Т. 2. С. 107–121.

280. Перечень террористов и экстремистов : сайт Федеральной службы финансового мониторинга Российской Федерации. URL: https://www.fedsfm.ru/documents/terrorists-catalog-portal-act (дата обращения: 25.12.2023).

281. Применение методов машинного обучения и отбора признаков на основе генетического алгоритма в решении задачи определения автора русскоязычного текста для кибербезопасности / А.В. Куртукова, А.С. Романов, А.М. Федотова, А.А. Шелупанов // Доклады ТУСУР. 2022. Т. 25, № 1. С. 79–85.

282. Реестр иностранных агентов. URL: https://minjust.gov.ru/ru/activity/directions/998/ (дата обращения: 29.06.2024).

283. Реестр информационных ресурсов иностранных агентов, доступ к которым ограничен на территории Российской Федерации. URL: https://426-fz.rkn.gov.ru/ (дата обращения: 29.06.2024).

284. Резанова З.И., Романов А.С., Мещеряков Р.В. Задачи авторской атрибуции текста в аспекте гендерной принадлежности (к проблеме междисциплинарного взаимодействия лингвистики и информатики) // Вестник Томского государственного университета. 2013. № 370. С. 24–28.

285. Резанова З.И., Романов А.С., Мещеряков Р.В. О выборе признаков текста, релевантных в автороведческой экспертной деятельности // Вестник ТГУ. Филология. Томск, 2013. № 6(26). С. 38–52.

286. Компьютерная обработка текстов при помощи ИС «СМАЛТ» / А.А. Рогов, Ю.В. Сидоров, А.В. Король, А.И. Солопова // Проблемы развития гуманитарной науки на Северо-западе России: опыт, традиции, инновации: материалы научной конференции. Петрозаводск, 2004. Т. 1. С. 122–124.

287. Проблема атрибуции в журналах «Время», «Эпоха» и еженедельнике «Гражданин» / А.А. Рогов, Р.В. Абрамов, Д.Д. Бучнева [и др.] // Издательство «Острова». 2021. 391 с.

288. Анализ тональности текстов с использованием методов машинного обучения (Sentiment Analysis of Text Using Machine Learning Techniques) / А.С. Романов, М.И. Васильева, А.В. Куртукова, Р.В. Мещеряков // Proceedings of the R. Piotrowski's Readings in Language Engineering and Applied Linguistics. Saint Petersburg, Russia, November 27, 2017. P. 86–95.

289. Романов А.С. Методика и программный комплекс для идентификации автора неизвестного текста: дис. … канд. техн. наук: 05.13.18 / А.С. Романов.

Томский государственный университет систем управления и радиоэлектроники. 2010. Т. 1. 149 с.

290. Романов А.С. Методология идентификации автора текста для решения задач информационной безопасности // Вопросы кибербезопасности. М. : НПО «Эшелон», 2024. № 3(61). С. 120–128.

291. Романов А.С. Методы отбора признаков в задаче определения авторства в контексте кибербезопасности // Моделирование, оптимизация и информационные технологии. 2024. Т. 12, № 1. URL: https://moitvivt.ru/ru/journal/pdf?id=1489.

292. Романов А.С. Модификация метода накопительных сумм для проверки однородности текста и выявления плагиата // Электронные средства и системы управления: материалы докладов IX Международной научно-практической конференции. Ч. 2. Томск: В-Спектр, 2013. С. 30–38.

293. Романов А.С. Идентификация автора текста для открытого множества кандидатов в контексте кибербезопасности // Моделирование, оптимизация и информационные технологии. 2024. Т. 12, № 1. URL: https://moitvivt.ru/ru/journal/pdf?id=1510.

294. Романов А.С. Методика идентификации автора текста на основе аппарата опорных векторов // Доклады ТУСУР. Томск, 2009. Ч. 2. № 1(19). С. 36–42.

295. Романов А.С., Резанова З.И., Мещеряков Р.В. Методика проверки однородности текста и выявления плагиата на основе метода опорных векторов и фильтра быстрой корреляции // Доклады ТУСУР. Томск, 2014. № 2(32). С. 264–269.

296. Романов А.С. Методика семантической кластеризации для выявления признаков экстремизма в текстовой информации // Доклады ТУСУР. Томск, 2024. Т. 27, № 4. С. 141–149.

297. Романов А.С. Модель базы данных для хранения текстов и их характеристик // Доклады ТУСУР. Томск, 2008. № 1(17). С. 70–73.

298. Романов А.С., Шелупанов А.А., Бондарчук С.С. Обобщенная методика идентификации автора неизвестного текста // Доклады ТУСУР. Томск, 2010. Ч. 1. № 1(21). С. 108–112.

299. Романов А.С., Шелупанов А.А., Мещеряков Р.В. Разработка и исследование математических моделей, методик и программных средств информационных процессов при идентификации автора текста. Томск, В-Спектр. 2011. 188 с.

300. Романов А.С. Структура программного комплекса для исследования подходов к идентификации авторства текстов // Доклады ТУСУР. 2008. Ч. 1. № 2(18). С. 106–109.

301. Рубцова Ю.В. Методы и алгоритмы построения информационных систем для классификации текстов социальных сетей по тональности: дис. канд. техн. наук: 05.13.19 / Ю.В. Рубцова. Федеральное государственное бюджетное учреждение науки Институт систем информатики им. А.П. Ершова Сибирского отделения Российской академии наук. Новосибирск, 2020. 141 с.

302. Рыжкова Е.С. Проблемы автороведческой классификационной экспертизы текстов интернет-коммуникации // Филологические науки. Вопросы теории и практики. 2015. № 6-1. С. 130–133.

303. Сайт ficbook.net. URL: https://ficbook.net/ (дата обращения: 20.06.2022).

304. Сайт github.com. URL: https://github.com/ (дата обращения: 20.06.2022).

305. Сбоев А.Г. Нейросетевое моделирование и машинное обучение на основе экспериментальных и наблюдательных данных: дис. … д-ра техн. наук: 05.13.18 / А.Г. Сбоев. Национальный исследовательский центр «Курчатовский институт». М., 2021. 389 с.

306. Сборник работ одаренных детей. URL: http://sykt-uo.ru/page-215.html (дата обращения: 30.06.2024).

307. Свидетельство о государственной регистрации базы данных № 2022622299. Российская Федерация. База данных естественно- и искусственно-языковых текстов для определения авторства : № 2022622243 : заявлено 16.09.2022 : опубликовано 21.09.2023 / Куртукова А.В., Романов А.С., Федотова А.М., Шелупанов А.А. Зарегистрировано в реестре баз данных.

308. Свидетельство о государственной регистрации базы данных № 2023623911. Российская Федерация. База данных исходных и

дизассемблированных кодов для идентификации авторства программ : № 2023623657 : заявлено 01.112023 : опубликовано 13.11.2023 / Куртукова А.В., Романов А.С., Федотова А.М., Шелупанов А.А. Зарегистрировано в реестре баз данных.

309. Свидетельство о государственной регистрации базы данных № 2024624287. Российская Федерация. База данных естественных текстов для идентификации пола и гендера автора текста : № 2024624158 : заявлено 05.10.2024 : опубликовано 14.10.2024 / Куртукова А.В., Романов А.С., Федотова А.М., Шелупанов А.А. Зарегистрировано в реестре баз данных.

310. Свидетельство о государственной регистрации программы для ЭВМ № 2020666001. Российская Федерация. Программа для тонального анализа текста «Moodorrude» : № 2020665103 : заявлено 26.11.2020 : опубликовано 03.12.2020 / Романов А.С., Васильева М.И., Шилов Л.С., Куртукова А.В., Шелупанов А.А. Зарегистрировано в реестре программ для ЭВМ.

311. Свидетельство о государственной регистрации программы для ЭВМ № 2022616780. Российская Федерация. Чат-бот для определения возраста автора текстового сообщения «Age Detector» : № 2022616248 : заявлено 12.04.2022 : опубликовано 15.04.2022 / Романов А.С., Соболев А.А., Федотова А.М., Куртукова А.В., Шелупанов А.А. Зарегистрировано в реестре программ для ЭВМ.

312. Свидетельство о государственной регистрации программы для ЭВМ № 2021681140. Российская Федерация. Система для идентификации автора исходного кода программы «CoDEtective» : № 2021667210 : заявлено 26.10.2021 : опубликовано 17.12.2021 / Куртукова А.В., Романов А.С. Зарегистрировано в реестре программ для ЭВМ.

313. Свидетельство о государственной регистрации программы для ЭВМ № 2022667584. Российская Федерация. Программный комплекс для получения информативных признаков автора текста «PyAuthorship» : № 2022667003 : заявлено 19.09.2022 : опубликовано 22.09.2022 / А.С. Романов, А.В. Куртукова, А.М. Федотова, А.А. Шелупанов. Зарегистрировано в реестре программ для ЭВМ.

314. Свидетельство о государственной регистрации программы для ЭВМ № 2023685210. Российская Федерация. Программный комплекс для идентификации запрещенных законодательством РФ и деструктивных текстов «PyDestructiveDetector» : № 2023682992 : заявлено 1.11.2023 : опубликовано 23.11.2023 / Куртукова А.В., Романов А.С., Федотова А.М., Шелупанов А.А.. Зарегистрировано в реестре программ для ЭВМ.

315. Свидетельство о государственной регистрации программы для ЭВМ № 2024685146. Российская Федерация. Программный комплекс для сбора данных о поле и гендере автора естественно-языкового текста : № 2024683353 : заявлено 10.10.2024 : опубликовано 24.10.2024 / Куртукова А.В., Романов А.С., Федотова А.М., Шелупанов А.А. Зарегистрировано в реестре программ для ЭВМ.

316. Свидетельство о государственной регистрации программы для ЭВМ № 2025615680. Российская Федерация. Программный комплекс для определения пола и гендера автора естественно-языкового текста «ProGender» : №2025613057 заявлено 20.02.2025 : опубликовано 06.03.2025 / Куртукова А.В., Романов А.С., Федотова А.М., Шелупанов А.А. Зарегистрировано в реестре программ для ЭВМ.

317. Свидетельство о регистрации электронного ресурса № 151146 Российская Федерация. Программная система для идентификации автора письменной речи «Автороведъ» : зарегистрировано 24.12.2009 : выдано 14.01.2010 / Романов А.С.. Зарегистрировано в Объединенном фонде электронных ресурсов «Наука и образование».

318. Созинова И.С., Романов А.С., Мещеряков Р.В. Определение поискового спама с использованием метода опорных векторов // Труды СПИИРАН. 2014. Вып. 36. С. 78–91.

319. Соколова Т.П. Проблемы экспертной идентификации в судебном автороведении // Вестник Университета имени О.Е. Кутафина. 2022. № 2(90). С. 67–76.

320. Стремоухов В.Д. Модель и метод анализа схожести и определения авторства вредоносного кода: дис. … канд. техн. наук: 05.13.19 / В.Д. Стремоухов. НИУ ИТМО, Санкт-Петербург. 2013. 95 с.

321. Суркова А.С. Идентификация авторства текстов на основе информационных портретов // Вестник Нижегородского университета им. Н.И. Лобачевского. 2014. № 3-1. С. 145–149.

322. Сухопаров М.Е. Методика идентификации пользователей порталов сети интернет на основе методов математической лингвистики: дис. … канд. техн. наук: 05.13.19 / М.Е. Сухопаров. Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, Санкт-Петербург. 2015. 108 с.

323. Тинькофф Журнал. URL: https://journal.tinkoff.ru/neuro-diploma/ (дата обращения: 29.12.23).

324. Ульченко А.Н. Идентификация авторства на основе векторизации лирики А.А. Ахматовой и М.И. Цветаевой психофизиологическими методами // Комплексные исследования детства. 2023. Т. 5, № 1. С. 4–13.

325. Федеральный закон «О контроле за деятельностью лиц, находящихся под иностранным влиянием» от 14.07.2022 № 255-ФЗ. URL: https://www.consultant.ru/document/cons_doc_LAW_421788/ (дата обращения: 29.06.2024).

326. Федеральный закон «Уголовный кодекс Российской Федерации» от 13.06.1996 (ред. от 25.12.2023) № 63-ФЗ. URL: http://pravo.gov.ru/ proxy/ips/?docbody&nd=102041891 (дата обращения: 24.05.2024).

327. Федеральный закон «О противодействии экстремистской деятельности» (с изменениями и дополнениями) от 25 июля 2002 г. № 114-ФЗ: сайт системы ГАРАНТ. URL: https://base.garant.ru/ 12127578/1cafb24d049dcd1e7707a22d98e 9858f/ (дата обращения: 25.12.2023).

328. Фоменко В.П., Фоменко Т.Г. Авторский инвариант русских литературных текстов // А.Т. Фоменко Новая хронология Греции: Античность в средневековье. М. : Изд-во МГУб, 1996. Т. 2. С. 768–820.

329. Хабр. За кулисами интеллекта ChatGPT: рассказ о том, как определяют тексты, созданные ИИ . URL: https://habr.com/ru/companies/antiplagiat/articles/ 728112/ (дата обращения: 29.12.23).

330. Хазова А.Б. Автоматическое определение половой принадлежности автора текста: феномен русской женской прозы // Вестник Новосибирского государственного университета. Серия: Лингвистика и межкультурная коммуникация. Новосибирск, 2020. Т. 18, № 1. С. 22–32.

331. Хмелев Д.В. Распознавание автора текста с использованием цепей А.А. Маркова // Вестник МГУ. Серия: Филология. 2000. № 2. С. 115–126.

332. Ходашинский И.А. Методы, модели, алгоритмы, инструментальные средства построения имитационно-лингвистических систем: дис. … д-ра техн. наук: 05.13.18 / И.А. Ходашинский. Томский государственный университет систем управления и радиоэлектроники. Томск, 2004. 357 с.

333. Чащин С.В. Применение методов машинного обучения с учителем для атрибуции текста: подходы и промежуточные результаты при идентификации авторов русскоязычных текстов // Вопросы криминологии, криминалистики и судебной экспертизы. 2018. № 1. С. 139–147.

334. Шевелев О.Г. Разработка и исследование алгоритмов сравнения стилей текстовых произведений: дис. ... канд. техн. наук : 05.13.18 / О.Г. Шевелев. Томский государственный университет систем управления и радиоэлектроники. Томск, 2006. 176 с.

ПРИЛОЖЕНИЕ А

СРАВНЕНИЕ ОСНОВНЫХ РАБОТ В ОБЛАСТИ АНАЛИЗА ТЕКСТОВ ДЛЯ ЗАДАЧ КИБЕРБЕЗОПАСНОСТИ

Таблица А.1 – Сравнение основных работ в области идентификации авторства текста

Метод или подход	Основные работы и авторы	Корпус и количество авторов	Признаки текста	Используемый объем одного текста, число образцов в классе	Различительная способность – точность методов
Методы статистического анализа, русский язык					
Взаимная энтропия	А.С. Суркова [321]	Художественные тексты XIX–XX веков, 20 авторов	Символьные *n*-граммы	Не указано	20 авторов – 88%
Деревья решений	А.Р. Дубовик [246]	Эссе в разных стилях (художественные, публицистические, научные, официально-деловые), 4 класса	Частотные распределения частей речи и синт. конструкций, признаки А.Ф. Журавлева и М.А. Марусенко	От 10 до 20 тыс. словоупотреблений / 65 текстов	4 автора – 57%
k-means	В.В. Исаченко [250]	Научные тексты на русском языке из базы Springer, 25 авторов	Word2vec, TF-IDF, частотные распределения символьных *n*-грамм, метаданные	Не указано / от 2 до 10 текстов	5 авторов – 95%, 25 авторов – 79%
	М.Ю. Кислицына, Ю.Н. Орлов [253]	Художественные тексты, 10 авторов	Символьные *n*-граммы	От 10 тыс. знаков / 5 и более текстов	10 авторов – 73%
Two-Block Projection to Latent Structure	А.Н. Ульченко [324]	Лирика Цветаевой и Ахматовой, 2 автора, 310 текстов	Частотные признаки	Не указано / 155 текстов	2 автора – 81%
Вычисление абсолютной разницы векторов	О.М. Атаева и др. [233]	Около 5000 авторов научных статей	Конкорданс словарь, словарь терминов, обратная частота	Не указано	2 автора – 85%
Методы статистического анализа, англ. и др. языки					
Метод базового профиля и дельта-метод	S. Ishihara, M. Carne [109]	1100 текстов на английском, 11 авторов	Вероятностные признаки	Не указано / 100 текстов	11 авторов – 91%
	R. Gorman [90]	Аннотации зависимостей, опубликованных в Древнегреческом и Латинском банке зависимостей, 13 авторов	Синтаксические и частеречные признаки, отбор признаков	50 слов / не указано	10 авторов – 89%, 13 авторов – 84%
Методы машинного обучения, русский язык					
SVM	Е. Пимонова и др. [171]	30 авторов, 215 текстов, написанных в период XVIII – XXI веков	Doc2Vec	От 350 предложений / 50 текстов	2 автора – 98%, 5 авторов – 89%
	М. Макарова [141]	Билингвальные русско-французские писатели, 10 авторов	Распределения суффиксов и прилагательных	500–1000 слов / 1–10 текстов	10 авторов – 85%
	Т.П. Соколова [319]	Короткие сочинения 20 человек	Частотные характеристики	Не указано	20 авторов – 59%
	П. Паничева [164]	RusIdioStyle, 5 авторов	*N*-граммы, частотный анализ частей речи	Средняя длина – 137 слов / 24 текста	5 авторов – 88%
KNN	С.В. Чащин [333]	Художественные тексты, 3 автора	Частотные распределения	5500 слов / не указано	3 автора – 87%

			местоимений		
LSA	А.С. Коляда, В.Д. Гогунский [254]	Научные статьи, 10 авторов	Ключевые слова	Не указано	60%
Методы машинного обучения, англ. и др. языки					
SVM, открытая атрибуция	M. Bartelds и др. [29]	Любительские тексты на англ., 9 авторов	Частоты биграмм знаков пунктуации, частей речи, *n*-грамм символов	500-1000 символов / 7 текстов	9 авторов – 87%
	F. Johansson и др. [113]	PAN-2014 Authorship identification, 2 класса	Стилометрические признаки, *n*-граммы символов, частоты регистра символов	Не указано / 50 текстов	2 автора – 98%
	K. Nishiyama и др. [157]	Японские и английские тексты, 5 авторов	Семантические и частотные признаки, BOW	13000 / 25–30 текстов, не указано	5 авторов – 96%
Ансамбль CAO и Манхэттенского расстояния	O. Halvani и др. [96]	PAN-2020 Authorship identification, 2 автора	Частоты слов, служебных частей речи, распределение глаголов по временам	21000 символов / 50 текстов	2 автора – 80%
LR, многослойный перцептрон	J. Weerasinghe и др. [215]		Стилометрические признаки		2 автора – 88%
General Impostors, Ranking-Based Impostors	N. Potha [176]	Кросс-тематические тексты на английском, 2 автора	Ранжирование частотных признаков	2700 слов / не указано	2 автора – 71-98%
SVM	H. Gómez-Adorno и др. [70]	Статьи газеты «The Guardian» 1999-2009, 13 авторов	Doc2Vec, *n*-граммы символов и частей речи (от 1 до 5)	1600-3000 символов / не указано	2 автора – 97%, 10 авторов – 90%
	S. Hedegaard, J. G. Simonsen [99]	The Federalist Papers, 71 статья 5 авторов Русская классика XIX вв., переведенная на англ. (30 оригинальных текстов, 4 автора, 12 переводчиков)	Семантические фреймы, 400 признаков	200–33000 символов / 7–10 текстов	Переведенные тексты, 4 автора – 58% Оригинальные, 5 авторов – 98%
LR	J. Grieve [94]	The Telegraph Columnist Corpus, 40 авторов	Частоты *n*-грамм слов	Не указано / 10 текстов	40 авторов – 69%
Семантические фреймы	R. Hinh и др. [101]	Extended-Brennan-Greenstadt corpus, 11 авторов	Индексы удобочитаемости	6500 слов / 15–22 текста	11 авторов – 82%
АСД	J. Patchala [167]	News Articles' Dataset (10 авторов и 50 текстов), Reuter 50-50 Dataset (50 авторов и 5000 текстов) и Blogs Dataset (100 авторов и 1000 текстов)	2000 *n*-грамм символов	200–800 слов / от 50 до 100 текстов	News Articles' Dataset, 10 авторов – 65%, Reuter 50-50, 50 авторов – 76%, Blogs Dataset, 10 авторов – 90%
Методы глубокого обучения, русский язык					

LSTM	М.М. Батурин и др. [235]	Художественные тексты писателей-классиков, 19 авторов, 1500 текстов	Fuse-функции, частоты частей речи, синтаксическая рекуррентная НС, словарь уникальных слов	100 предложений / 1 текст	19 авторов – 68%
Графовые НС, ранговый критерий Краскела-Уоллиса, рекуррентная НС, LSTM, модель Transformer, дерево решений, SVM, RF	Н.Д. Москин [274, 287]	500 текстов неустановленного авторства, 428 песен, 210 стихов и песен А.К. Толстого, 20 стихов и песен С.А. Есенина, 46, приписываемых А.С. Пушкину, 63 фрагмента текста Ф.М. Достоевского	Непараметрический ранговый критерий Краскела-Уоллиса	Не менее 1000 слов/ от 2 до 50 текстов	2 автора – 97%, 5 авторов – 90%, 50 авторов – 43%
SVM с отбором признаков (ген. алгоритм, регуляризационный отбор), GRU+CNN, открытая атрибуция, верификация One-Class SVM	**А.С. Романов [20, 21, 185, 66, 291]**	**Художественные тексты – 50 авторов, 776 текстов, любительские тексты – 69 авторов, 6569 текстов**	***N*-граммы символов, распределение частей речи и знаков пунктуации, частотный словарь, отбор генетическим алгоритмом, регуляризационный отбор**	**15000 символов / 3 образца**	**Закрытая атрибуция: 2 автора – 99%, 5 – 98%, 10 – 94%, 20 – 88%, 50 – 57%. Открытая атрибуция: 2+null – 96%, 4+null – 92%, 9+null – 85%, 19+null – 67%. Верификация: худ. – 99,2%, люб. - 98,9%.**
Методы глубокого обучения, англ. и др. языки					
LSTM	B. Boenninghoff [32, 33]	PAN-2020 Authorship identification, 2 автора	Статистическое моделирование	21000 символов / 50 текстов	2 автора – 94%
Transformer	M. Najafi и др. [152]	PAN-2018, 1 класс	Частеречные распределения	350 слов / не указано	1 автор – 86%
Ансамбль лексической и синтаксической подсетей	F. Jafariakinabad и др. [111]	LAMBADA, 2662 текста, 3 класса	BOW	300-1000 символов / не указано	3 автора – 92%
fastText	T. Van Tussenbroek [208]	Субтитры 236 серий телесериала «Друзья», 6 авторов	TF-IDF	300 слов / не указано	6 авторов – 36%
Graph-based Siamese Network	D. Embarcadero-Ruiz и др. [71]	PAN 2018, 4 автора	Взвешенные признаки графовой подсети	345 предложений / не указано	4 автора – 95%
CNN	D. Boumber [38]	400 научных публикаций в области машинного обучения, 20 авторов	GloVe, длина слова, предложения	Не указано / 20 текстов	20 авторов – 65%

Multi-Channel Self-Attention Network, BiLSTM	H. Wu и др. [218]	CCAT10 (10 авторов), CCAT50 (50 авторов) и IMDb62 (62 автора)	Синтаксические признаки	1780–3100 символов / от 2 до 20 текстов	10 авторов –93%, 50 авторов – 83%, 62 автора – 69%
Topic Drift Model, Gaussian Mixture Neural Topic Model	M. Yang и др. [222]	IMDb62, 20 авторов	Евклидово расстояние, частотные признаки	Не указано / от 350 до 1000 текстов	20 авторов – 52-93%
Ансамбль XGBoost, RF и MLP, Multi-depth DistilBERT	A. Abbasi и др. [2]	143000 новостных статей 16408 авторов	Count Vectorizer и TF-IDF	Не указано / 350 статей	97% – 10 авторов, 79% – 20 авторов
GPT-4 Turbo, LIP	B. Huang и др. [106]	Blog Authorship Corpus (19000 авторов), Enron Email Corpus (около 150 авторов)	Лингвистические признаки: модальные глаголы, пунктуация, редкие слова, фразовые глаголы, аффиксы, юмор, сарказм, опечатки, пропуски и др.	40–79 токенов / 1 текст	Электронная почта: 10 авторов –88,9%, 20 авторов – 77,2%. Посты в соц. сетях: 10 авторов – 86,7%, 20 авторов – 60,5%.
RoBERT	M. Nitu и др. [158]	Рассказы на румынском языке, 19 авторов	Частоты слов, длин предложений и синтаксических структур, эмбеддинги RoBERT, отбор признаков с использованием среднего ранга Краскела-Уоллиса	Не указано / 60–66 текстов	19 авторов – 87%

Таблица А.2 – Сравнение основных работ в области идентификации пола и возраста автора текста

Метод или подход	**Основные работы и авторы**	**Корпус, количество текстов, число классов**	**Признаки текста**	**Используемый объем одного текста, число образцов для обучения в классе**	**Различительная способность – точность методов**
Определение пола и возраста автора текста, русский язык					
SVM	А.К. Ковалёв и др. (пол и возраст) [268]	487 эссе взрослых и студентов, группы 10–20, 21–30, 31–40 лет, пол – 2 класса	Уровень агрессивности по опроснику Басса-Перри	0,5–1 страница / 220 (пол), 130 (возраст)	Определение пола – 68%, Определение возраста – 55%
	И. Марков (пол) [144]	Twitter* (400 текстов), Facebook** (228 текстов), отзывы (776 текстов), 2 класса	*N*-граммы символов	Twitter* – 280, Facebook** – 670, отзывы – 280 символов / Twitter* – 320, Facebook** – 180 и отзывы – 650 текстов	Определение пола: Twitter* – 93%, Facebook** – 68%, Отзывы – 62%. Средняя – 65%.

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.
** Деятельность организации Meta Platforms Inc, ее продукта Facebook запрещена в Российской Федерации.

LSTM и BiLSTM	R. Bhargava и др. (пол) [31]	600 тематических эссе на русском, 2 класса	Неявное извлечение признаков НС	Не указано / 300 текстов	Определение пола – 78%
Анализ частеречных распределений	И.В. Огорелков (пол) [277]	1000 текстов пользователей социальных сетей, 2 класса	Смысловые, текстологические, языковые, психолингвистические, лексические, синтаксические, психолингвистические признаки	500 символов / 500 текстов	–
NB, RF, KNN	А.Б. Хазова (пол) [330]	Русскоязычная художественная проза 1960–2000 гг, 87 текстов, 2 класса	Частоты знаков пунктуации и частей речи	Не указано / 42 текста	Определение пола – 70%
RF, AdaBoost, деревья решений и KNN	Д.А. Галкина (пол) [242]	146 текстов (79 художественных, 67 других жанров), 2 класса	Частотные распределения местоимений	Не указано / 70–76 текстов	Определение пола – 85%
SyntGraphLSTM, эмбеддингах на основе символов и извлеченных из Сиамской НС	А.Г. Сбоев (пол и возраст) [305]	1850 текстов, написанных в рамках эксперимента и 41624 из социальных сетей (возрастные группы 18–23, 24–29, и старше 30 лет), пол – 2 класса. Средняя длина 300 символов. 4332 текста с искажением пола, 13632 с искажения возраста	Частоты последовательностей символов от триграмм до октаграмм, формирование признаков с учетом частей речи и эмоциональной составляющей;	От 100 до 1000 символов / от 1000 текстов	Определение пола – 86%, Определение возраста – 80%
fastText, VGG-Face	**А.С. Романов (возраст) [64, 266, 53, 187]**	**«ВКонтакте»: 2994 автора и 202892 текста, 3 возрастные категории, 3 класса (до 18, от 18 до 21, старше 21 года)**	**Метрики удобочитаемости**	**200–300 символов / 68000 текстов**	**Определение возраста – 82% для 2 классов, 63% для трех классов**
SVM, CNN, BERT, семантическая кластеризация, сглаживание вероятностей	**А.С. Романов (пол)**	**Telegram, X* – 706178 текста, 2 класса**	**Сглаженные N-граммы символов, ключевые семантические слова**	**От 50 символов / 153315 текстов**	**Определение пола – 92%**
Определение пола и возраста автора текста, англ. и др. языки					
NB, SVM, многослойный персептрон, дерево решений и их ансамбль	D.I. Hernández (пол и возраст) [100]	12000 коротких публикаций из общедоступного корпуса PAN 13 Training Corpus for Author Profiling Task, 3 класса возраст, пол – 2 класса	WordNet: Similarity, BOW, коэффициент Жаккара	Не указано / 6000 текстов (пол), 1400 текстов (возраст)	Определение пола – 52%, Определение возраста – 48%

Neural Machine Translation, LSTM	E. Vanmassenhove и др. (пол) [209]	Europarl Corpus, 20 языковых пар, пол – 2 класса	Морфологические признаки, синтаксические конструкции	Не указано	BLEU: EN-FR: +1,44 EN-IT: +0,29
RF	P. Vashisth и др. (пол) [212]	PAN 2019 Author Profiling Shared Task, пол – 2 класса, проверка искусственного текста – 2 класса	17 признаков на основе стилометрии (частоты специальных символов, знаков пунктуации, коротких слов до 5 символов, длинных слов от 11 символов) и метаданных (число действий с твитом – количество ретвитов, добавлений в избранное, комментариев)	До 140 символов / 100 текстов на 1 человека	Определение пола – 78%,
Hinge-loss Markov Random Fields (HL-MRFs)	F. Golnoosh и др. (пол и возраст) [76]	Facebook*: 5670 пользователей, 724948 текстов, пол – 2 класса, возраст – 2 класса (до 23 лет и старше 23 лет)	Текстовые (статус), визуальные (фото профиля), социальные связи (отметки «нравится»)	Не указано	Определение пола – 81%, Определение возраста – 88%
Глубокие сети усреднения	M. Franco-Salvador и др. [82]	PAN 2017 Author Profiling Shared Task, пол – 2 класса	Извлечение подстрок, *n*-граммы	До 140 символов / 100 текстов на человека	Определение пола – 88%
SVM	J. Pizarro и др. (пол) [173]	PAN-2019 (англ. тексты), 4000 текстов, пол – 2 класса	TF-IDF, *n*-граммы символов	Не указано / 2000 текстов	Определение пола – 84%
	F. Rangel и др. (пол и возраст) [182]	PAN-AP-13, тексты на испанском, пол – 2 класса, возраст – 3 класса (10–20, 20–30, 30–40 лет)	TF-IDF, *n*-граммы символов	Более 1000 слов в тексте каждого автора / не указано	Определение пола – 63%, Определение возраста – 63%
	S. Daneshvar (пол) [58]	PAN 2018 Author Profiling Task, English, Spanish, Arabic, Gender Classes, 2 класса	*N*-граммы слов и символов, LSA	Не указано / обучение – тексты 13332 пользователей, валидация и тестирование – по 20% (4444 пользователей)	Определение пола – 82%
LR	S. Mechti и др. (пол и возраст) [147]	PAN за 2014–2016, 262040 текстов, возраст – 3 класса, пол – 2 класса	Doc2Vec, 200 частотных признаков	Не указано / 236600 текстов (пол), 78666 текстов (возраст)	Определение пола – 76%, Определение возраста – 79%
ReliefF, RF, NB, SVM	D. Nguyen и др. (возраст) [19, 156]	19377 твитов и шесть классов пользователей (14-19, 20-24, 25-34, 35-44, 45-59, 60+ лет)	Частотные признаки (части речи, пунктуация)	До 280 символов / 2500–4000 текстов	Определение возраста – 61%

* Деятельность организации Meta Platforms Inc, ее продукта Facebook запрещена в Российской Федерации.

RoBERTa	I. Lanza-Cruz и др. (возраст) [129]	14343 текста, три класса (до 25, от 26 до 40 и старше 40 лет)	fastText	Не указано / 4781 текст	Определение возраста – 88%
MLP	J.S. Alowibdi [10]	6000 твитов, созданных ИИ и человеком, 4 класса	TF-IDF, отбор признаков с применением теста Хи-квадрат	До 280 символов / 1500 текстов	4 класса – 90%
GPT2	V. Younkin и др. [226]	8222 художественных текста, 2 класса	N-граммы слов, контекстные вложения	Не указано / 4111 текст	2 класса – 92,5%
Определение признаков ЛГБТ в тексте, русский язык**					
LR	C.Y. Park и др. [166]	Страницы ЛГБТ-активистов на русском, английском и испанском, 10000 образцов, 2 класса	Символьные *n*-граммы	200-500 символов / 5000 текстов	2 класса – 83%
BERT	N. Babakov и др. [24]	Сгенерированные тексты на русском, 124597 образцов, 2 класса	Символьные *n*-граммы	100-600 слов / 62250 текстов	2 класса – 80%
SVM, CNN, BERT, семантическая кластеризация, сглаживание вероятностей	**А.С. Романов**	**Telegram, X – 706178 текста, 2–6 классов**	***N*-граммы символов, ключевые семантические слова, сглаживание Катца**	**От 50 символов / от 36166 до 336316 текстов в зависимости от эксперимента, сбалансированные наборы**	**2 класса – 93%, 3 класса – 75–80%, 4 класса – 85%, 6 классов – 68%**
Определение признаков ЛГБТ в тексте, англ. и др. языки**					
NB, SVM с линейным ядром, LR и XGBoost	A. Соесанто и др. [201]	Мнения о представителях ЛГБТ**, 20000 текстов, 2 класса	TF-IDF	Не указано / 10000 текстов	72 класса – 1%
RF	T.E. Workman и др. [217]	Электронные медицинские карты (977481 запись о 153 212 пациентах), 2 класса	Ключевые слова	Не указано / 1582 текста	2 класса – 88%
NB	A. Karami и др. [119]	Twitte*r, 16000 аккаунтов, 2 класса	Метаданные постов, число публикаций, число подписчиков и подписок, частоты символьных униграмм и цифр	До 280 символов / 10000–20000 текстов	2 класса – 88%
XLM-RoBERTa	D. Manikandan и др. [143]	990 комментариев, 2 класса	BERTVectorizer	1–20 предложений / 495 текстов	2 класса – 93%

** Запрещенная в Российской Федерации экстремистская организация.

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

Модели компьютерного зрения для определения возраста автора					
Gated Residual Attention Network	A. Garain и др. [84]	FG-Net (1002 фото, 0–69 лет), AFAD (160000 фото, 15–40 лет), набор данных из Википедии (62328 фото, 5–80 лет), UTKFace (20000, 0–116 лет)	CNN, облако точек	От 1 до 5 фото каждого человека / 800 фото (FG-Net), 120000 фото (AFAD), 55000 фото (Википедия), 16000 фото (UTKFace)	Ошибка регрессионной модели от 1 до 10 лет
MobileNetV2	B.A. Kumar и др. [123]	20000 фото	Облако точек	От 3 до 5 фото каждого человека / 16000 фото	Ошибка – 2,8 лет

Таблица А.3 – Сравнение основных работ в области идентификации автора сообщений в сети Интернет и продленная аутентификация пользователей

Метод или подход	Основные работы и авторы	Корпус и количество авторов	Признаки текста	Используемый объем одного текста, число образцов в классе	Различительная способность – точность методов
Короткие сообщения соц. сетей и мессенджеров, русский язык					
SVM, NB, RF	А.А. Воробьева и др. [240]	Электронные текстовые сообщения длиной до 5000 символов	Частотные признаки, IP-адрес, версия операционной системы	2700–8200 символов / 25 текстов	10 авторов – 80%
Метод обнаружения компонентного профиля	М.Е. Сухопаров [322]	Короткие тексты на русском языке пользователей порталов mail.ru, drom.ru, ixbt.ru и awd.ru	Энтропия Шеннона, словарь В.А. Тузова	Средняя длина – 8 слов / 50–3000 текстов	Вероятность обнаружения атаки вероятности для систем мониторинга состояния информационной безопасности – 67%
RNN, CNN, LSTM	Y.V. Dmitrin и др. [68]	Habrahabr blog – 30 авторов, vk.com – 50 и 100 авторов, Echo.msk.ru 50 и 100 авторов	Триграммы и тетраграммы символов	100–500 слов / 300–500 текстов	50 авторов – 55%, 100 авторов – 50%
SVM с отбором признаков (ген. алгоритм, регуляризационный отбор), GRU+CNN; открытая атрибуция, верификация – One-Class SVM и GRU+CNN	**А.С. Романов [21, 66, 291]**	**«ВКонтакте» – 2962 авторов, 325892 текста, 6742 искусственных текста**	***N*-граммы символов, распределение частей речи и знаков пунктуации, частотный словарь, отбор генетическим алгоритмом, регуляризационный отбор**	**100–250 символов / 40 образцов**	**Закрытая атрибуция: 2 автора – 96%, 5 – 93%, 10 – 78%, 20 – 74%, 50 – 56% Открытая атрибуция: 2+null – 93,5%, 4+null – 86,9%, 9+null – 81,2%, 19+null – 69,8%, 49+null – 51,9%. Верифиикация – 97,8%**

Короткие сообщения соц. сетей и мессенджеров, англ. и др. языки					
XGBoost	S. Alterkavı и др. [13]	16124 твитов 20 пользователей Twitter*	Multiple-criteria decision analysis	До 140 символов / не указано	20 авторов – 94%
SVM	S. Adamovic и др. [23]	513 английских текстов, 5 авторов	124 признака (*n*-граммы, средняя длина текста, регистр букв, частоты стоп-слов и цифр)	315 слов / 50–100 текстов	5 авторов – 90%
fastText, SVM	H.A. Chowdhury и др. [50]	3 автора Интернет-блогов, 500000 слов	*N*-граммы от 1 до 4	300 символов / 50 текстов	3 автора – 88%
CNN-1 – униграмм-ная, CNN-2 – биграм-мная	S. Loaiza и др. [197]	50 авторов Twitter* на английском	Униграммы и биграммы символов	140 символов / 1000 символов,	20 авторов – 85%
Многопоточные RNN на символьном уровне	G. Barlas и др. [28]	Английские тексты 21 студента, 42 текста	Universal Language Model Fine-Tuning (ULMFiT)	Не указано / 2 текста	21 автор – 89%
DT, SVM, RF, NB	C. Deutsch, I. Paraboni [65]	Корпуса электронной почты Enron, набора данных HUMAINE и коллекции сообщений Twitter*	Частеречные распределения, частоты *n*-грамм слов	140-500 символов / не указано	10 авторов – 87%
TDRLM	X. Hu и др. [104]	Twitter-Foursquare: 45756 текстов, 5392 автора	Стилометрическое обучение, словарь тематических оценок	Не указано / 8 текстов	Задача верификации – 92,47%

Таблица А.4 – Сравнение основных работ в области идентификации автора исходного кода программы

Метод или подход	**Основные работы и авторы**	**Корпус и количество авторов**	**Признаки текста**	**Используемый объем одного исх. кода, число образцов в классе**	**Различительная способность – точность методов**
Source Code Author Profile	G. Frantzeskou и др. [83]	MacDonell C++ (6 авторов, 286 исходных кодов), NoCom Java (8 авторов, 107 исходных кодов)	Профили *n*-грамм на уровне байтов	206 и 122 символа / 5 и 22 исходных кода	MacDonell C++, 2 автора – 97%, NoCom Java, 2 автора – 94%
	W. Wisse и др. [216]	Java Script, 167 авторов,	АСД, *n*-граммы	7,7 КБ / 5–32 файла	10 авторов – 85%
Подсчет ключевых подстрок исходного кода и различных стилистических характеристик	S. Burrows и др. [42]	Исходные коды программ 1597 студентов технических специальностей на языке C	*N*-граммы, литералы, ключевые слова, операторы, функции, пробельные символы	650 строк / не указано	10 авторов – 77%

* Социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

НС обратного распространения	X. Yang и др. [223]	GitHub, Java, 40 авторов, 3022 файла	Метод оптимизации роя частиц	Не указано / 81 исходный код	40 авторов – 75–91%
LSTM и BiLSTM	B. Alsulami и др. [12]	70 программистов Python, 700 файлов, 10 разработчиков C++, 200 файлов	АСД	Не указано / C++: 20 файлов, Python: 10 файлов	Python, 25 авторов – 92%, C++, 10 авторов – 85%
Архивация, матрица переходных вероятностей цепи Маркова	В.Д. Стремоухов [320]	Delphi, C++, 10000 образцов	Степень сжатия, словарь над байтовым алфавитом	Не указано	Корректность обнаружения вредоносного кода при пороговом значении 0,7% – 97%
Нечеткие АСД, KNN, NB	T. Zia и др. [229]	Java, 153 файла, 9 авторов	Стилометрические признаки и признаки дефектов авторского кода, итераторы, длина строки, соотношение отступов к длине кода	1000 строк / 17 исходных кодов	9 авторов – 75%
RF, открытая атрибуция	A. Caliskan-Islam и др. [44, 43]	Google Code Jam (C++ и C, 250 и 1600 авторов)	АСД, *n*-граммы, статистические значения	Не указано / 6–9 файлов	2 автора (открытая атрибуция) – 97%, 10 авторов (закрытая атрибуция) – 98%
Статический и динамический анализ исходного кода, сиамская НС	N. Wang и др. [213]	Google Code Jam, 1159 Python-разработчиков, 80000 исходных кодов	АСД, лексические признаки и отступы, использование функций и импорта пакетов, использование памяти	Не указано	25 авторов – 75%
XAI, CNN, KNN	A. Bogdanova и др. [34]	Google Code Jam для языков C++, Java и Python	Лексические и структурные признаки, эмбеддинги BERT	800 символов и более / не указано	Попарное сравнение, средняя точность, C++ – 74%, Java – 77%, Python – 73%
PbNN и модифицированный RF	E. Bogomolov и др. [35]	C++ (1600 авторов, 14400 файлов), Python (70 авторов, 700 файлов), Java (40 авторов, 3021 файл)	АСД	Не указано / C++ – 9 файлов, Python – 10 файлов, Java – 11–712 файлов	Попарное сравнение, средняя точность, Python – 96%, Java – 99%, C++ – 95%
DNN, RF и C-SVM, v-SVM	P. Mahbub и др. [139]	Мультиязычный набор данных, 8 авторов	Статистические характеристики (длины и средние величины)	226 строк / 4–50 файлов	5 авторов – 87%
TFIDF_NN, CNN	A.V. Mandalam и др. [142]	AI-SOCO, C++ (1000 авторов, 100000 исходных кодов)	TF-IDF, Word2Vec	Не указано / 100 файлов	Попарное сравнение, средняя точность – 91%

RoPGen, RF, многоуровневые RNN	L. Zhen и др. [228]	C++ (204 автора, 1632 файла), Java (40 авторов, 2827 файлов), C (67 авторов, 2072 файла)	TF-IDF, признаки на уровне токена, функции и модуля	C++ – 74 строки кода, Java – 76 строк кода, C – 88 строк кода / не указано	19 авторов, C++ – 95%, Java – 42%, C – 65%. При атаке, направленной на имитацию стиля: C++ – 81%, C – 54%
Метод графа зависимости программы с глубоким обучением, SMOTE	F. Ullah и др. [206]	Google Code Jam, C++ (540 авторов), Java (320 авторов), C# (140 авторов)	Метод синтетической избыточной выборки меньшинства, TF-IDF	Не указано / 10-100 файлов	C++: 20 авторов – 77%, 100 авторов – 61%, Смешанный датасет: 1000 авторов – 68%
VGG, LSTM	P. Bora и др. [36]	GitHub Java файлов 20 авторов, 20000 обучающих образцов, 13263 тестовых образцов	Изображения кода, преобразованные в векторные представления с помощью VGG16, CNN и LSTM	В среднем 100 строк кода на файл, с использованием пересекающихся фрагментов кода / не указано	20 авторов – 85%
CLAVE	D. Álvarez-Fidalgo, F. Ortin [14]	GitHub, GCJ Python (819405 исходных кодов, 77703 авторов)	Стило-метрические признаки	1000 и более символов / 1 исходный код	Задача верификации: F1 – 93%, AUC – 0,97.
AuthAttLyzer-V2	B. Joshi, S.H. Khani, A. HabibiLashkari [114]	Codeforces C++ (24000 исходных кодов, 3000 авторов)	54 статических и динамических признаков	Не указано / 8 исходных кодов	Точность – 81,2%, полнота – 80,5%, F1 – 81,2% для 8 авторов.
CodeBERT, обфускация, стандарты кодирования, смешанные данные, коммиты, искусственно-сгенерированные исходные коды	**А.С. Романов [258, 124, 64, 53, 261]**	**GitHub (13 языков, 102908 исходных кодов, 383 автора), Google Code Jam**	**Структурные, лексические, нарушающие парадигмы программирования признаки**	**170 строк / 30 файлов**	**GitHub: C++ – 86% Java – 91% JS – 87% Python – 91% C – 92% C# – 93% Ruby – 90% PHP – 86% Swift – 85% Go – 84% Groovy – 91% Kotlin – 85% Perl – 84%. GCJ (средняя точность): C++ – 94% Java – 97% Python – 95%**

Таблица А.5 – Сравнение основных работ в области определения тональности и деструктивной и экстремистской направленности текста

Метод или подход	Основные работы и авторы	Корпус текстов	Признаки текста	Используемый объем одного текста, число образцов в классе	Различительная способность – точность методов
Определение тональности текста, русский язык					
Математическая модель на основе многоуровневых графов, соседи первого, второго и третьего порядка	А.О. Корней [256]	«КиноПоиск» (506 текстов) и набор отзывов о ресторанах из датасета SemEval-2016 (312 текстов)	Словарь математической энциклопедии (7500 словарных статей и более 2 миллионов слов), *n*-граммы символов	30–100 слов / 300–20000 текстов	3 класса – 69,5%
KNN, Random Forest, SVM	N. Chechneva [48]	41913 отзывов о товарах, «Яндекс.Маркет», 32 категории	Качественные и количественные прилагательные	Не указано / 750 предложений	2 класса – 80%
BERT	И.А. Андреев [230]	100 пользователей, имеющих профили в различных социальных сетях (всего 204 аккаунта)	Схожесть контактной информации профилей, лиц на фотографиях, места работы и учебы, наличие схожих сообщений со страниц профиля, расстояние Левенштейна, алгоритм шинглов, WordNet-Affect	200–1000 символов / 566–2100 текстов	7 классов – 83%
Схема взвешивания с линейной сложностью обработки	Ю.В. Рубцова [301]	77 миллионов сообщений пользователей социальной сети Twitter*) и датасет SemEval-2016	Словарь, Skip-gram и усредненные векторы слов из сообщений	Не указано / положительная коллекция – 150720 уникальных словоформ; нейтральная коллекция – 105239 уникальных словоформ; негативная коллекция – 191677 уникальных словоформ	2 класса – 93%, 3 класса – 90%
SVM, NB, fastText, KNN, DT, адаптивный бустинг, BERT	С.В. Вычегжанин [241]	Комментарии «ВКонтакте» по темам «Вакцинация детей», «Клонирование человека» и «ЕГЭ в школе»	DNRFAF, DSRFE и DECCV	Не указано	5 классов – 78%

* социальная сеть X (бывший Twitter) запрещена на территории Российской Федерации.

RNN и CNN	И. Гринин [244]	Корпус Юлии Рубцовой, 114991 положительных, 111923 отрицательных твитов, 17639674 неразмеченных	Word2Vec	Не указано / 114991 сообщение	2 класса – 79%
Определение тональности текста, англ. и др. языки					
NB, Random Forest, KNN.	S.K. Acharya [6]	96775 положительных и 22741 отрицательных отзывов	Count-Vectorizer, TF-IDF	Не указано / 2000 образцов	2 класса – 87%
Ансамбль: LR, многослойный перцептрон и SVM	Y. Görmez и др. [91]	IMDB, набор данных турецких фильмов, собранный с веб-страницы Beyazperde, и SentiRuEval-2017	TF-IDF, униграммы символов, skip-gram	Не указано / IMBDB: 25000 текстов, Beyazperde: 5313 текста, SentiRuEval-2017: 7059 позитивных, 3231 негативных, 10342 нейтральных текстов	Beyazperde, 3 класса – 89%, SentiRuEval-2017, 3 класса – 64%, IMBDB, 2 класса – 89%
CNN, LSTM, BiLSTM с механизмом внимания и развертывающая НС	D. Pan и др. [163]	Отзывы по тематике весеннего фестиваля в Китае и туризма	Word2Vec	До 300 слов / 3828 позитивных, 3957 негативных и 6989 нейтральных текста	3 класса – 78%
LLaMa и BloomZ, mBART и condBERT.	Д. Дементьева и др. [62]	PAN-2024, мультиязычная детоксикация текста. 30000, 2700 и 720 текстов на русском, украинском и испанском, соответственно	Фильтрация токсичных комментариев, по ключевым словам,	Не указано / 720–30000 в зависимости от языка	Style Transfer Accuracy 95%, 69% и 45% для русского, украинского и испанского
Определение деструктивной и экстремистской направленности в текстах, русский язык					
LinearSVC	Т.А. Литвинова [263, 135]	KavkazChat* (699981 текст, 7125 авторов)	Linguistic Inquiry and Word Count (LIWC), фильтрация по словарю	Не указано / от 50 текстов	20 классов – 49%
	V. Moshkin [149]	Социальные сети, 1010 документов, 2 класса (опасные и подозрительные посты)	Лексические признаки, лематизация, гиперонимы	Не указано / 190 опасных, 820 подозрительных постов	2 класса – 84%
Gradient Boosting	S. Mussiraliyeva и др. [151]	Экстремистские тексты (3000 слов) и тексты неэкстремистской религиозной или новостной направленности (15000 слов)	TF-IDF, Word2Vec	3000-15000 слов / не указано	2 класса – 90%
LR, RF	В.А. Минаев и др. [270, 272]	«ВКонтакте»	BOW, TF-IDF и Word2Vec	500–2000 символов / не указано	2 класса – 93%

* признан экстремистским в Российской Федерации.

NB	А. Капитанов и др. [118]	Dark Web, более 100000 текстов на русском, английском и китайском из запрещенных источников	Лексические, синтаксические и тональные признаки, включая частоту использования слов, кластеризацию и анализ тональности текста	500 символов / 400 текстов	2 класса – 88%
Алгоритм иерарх. кластеризации, LR с L2 регуляризацией, дерево решений типа CART, RF и ансамбль DTна основе XGBoost и LightGBM	И.В. Машечкин и др. [267, 269]	KavkazChat* (699981 текст, 7125 авторов)	Матрица термов веток текстовых сообщений	Не указано / от 1000 до 10000 текстов	10 кластеров экстремистских категорий – 86%
SVM, KNN, NB и RNN	А.А. Гончаров [243]	Неструктурированные тексты с заранее известными классами: классический ислам, радикальный ислам, нейтральные тексты	Word2Vec и TF-IDF	Не указано / 2100 текстов	2 класса – 87%
RF и градиентный бустинг	А.К. Ковалёв и др. [268, 122]	Эссе по одной тематике 487 испытуемых от 18 лет	Шкала агрессивности по опроснику Басса–Перри, PCA	0,5–1 страница / 160 текстов	3 класса – 55%
Гибридная модель (лингвистические правила и статистические данные) для присвоения веса различным типам текстовых ошибок	Ю.В. Давыдова [245]	Комментарии пользователей социальных сетей, 600000 слов	Анализ на уровне строк и словоформ, расстояние Левенштейна	Не указано / 300000 слов	2 класса – 93%
fastText, LR и MultinomialNB	В. Охапкин и др. [160]	Социальные сети, 1048576 сообщений	BOW, словарь WordNet-Affect	Не указано / 100 сообщений	2 класса – 64%
fastText	А. Исхакова и др. [110]	Pikabu, 12333 текстов, 3 класса (позитивный, нейтральный, токсичный)	Частота слов (униграммы, биграммы, триграммы)	Не указано, 3071 токсичных, 7023 нейтральных, 2239 позитивных	3 класса – 63%
BERT, GRU+CNN, SVM, семантическая кластеризация	**А.С. Романов [193]**	**Telegram (новостные каналы, экстремистские каналы, каналы иностранных агентов), ВКонтакте, KavkazChat*, 778397 текстов 11000 авторов**	**Эмбеддинги BERT**	**Определение автора деструктивного текста: 50–250 символов, / 50 образцов. Определение деструктивного**	**Определение автора деструктивного текста; Закрытая атрибуция: 2 автора – 95%, 5 – 91%, 10 – 80%,**

				контента: 50-250 символов / 15000-59000 текстов	**20 – 71%. Открытая атрибуция: 2+null авторов – 93%, 4+null авторов – 90%, 9+null авторов – 78%, 19+null авторов – 70%. Верификация – 90%. Определение деструктивного контента (2 класса) – 94%**
Определение деструктивной и экстремистской направленности в текстах, англ. и др. языки					
SVM	Л. Раналди и др. [181]	Форум, посвященный обсуждению вопросов, связанных с противозаконными организациями (91874 сообщений, 2082 активных пользователя)	TF-IDF	Не указано / 1400 текстов	2 класса – 85%
C4.5 и SVM	A. Abbasi и др. [1]	Арабские сообщения на форумах, связанных с известными противозаконными группами	Лекс., синт. и структурные признаки	Не указано / 20 текстов	2 класса – 90%
DistilBERT	M. Abdalsalam и др. [3]	180000 текстов, 7 классов	Эмбеддинги DistilBERT, *n*-граммы символов	200 слов / не указано	7 классов – 97%

Таблица А.6 – Сравнение основных работ в области методов атаки на методы идентификации автора

Метод или подход	**Основные работы и авторы**	**Корпус текстов**	**Признаки текста**	**Используемый объем одного текста, число образцов в классе**	**Различительная способность – точность методов**
Обфускация					
3 обфускатора, влияющих на эффективность 44 подходов верификации авторства	M. Поттхаст и др. [177]	PAN Obfuscation, 390 авторов, 10 000 текстов	Косинусное сходство, TF-IDF, TF-weighted	50 слов / не указано	90 классов – 47%
Mutant-X, функции кроссовера и приспособленности	A. Mahmood и др. [140]	Extended-Brennan-Greenstadt, 45 авторов	Признаки на уровне символов, слов и предложений, частоты уникальных слов	Среднее значение – 492 слова / не указано	5 авторов – 90%, 10 авторов – 70%
SVM03, метод Коппеля и подход Шлера	G. Kacmarcik и др. [117]	Federalist Papers, 85 документов, 3 автора	Выделение частотных признаков и их замена согласно словарю	1000 слов и более / 14–51 текст	3 класса – 83%

Маскирование, открытая атрибуция, Generative Adversarial Network, Сиамская НС	R. Balakrishnan и др. [26]	Reuters C50, 50 авторов, 2500 статей	Изменение *n*-грамм слов и символов, анонимизация на уровне предложений	10 слов и более / 50 текстов	Средние значения: BLEU – 0,61, BERTScore – 0,93
Имитация					
SyntGraphLSTM, сиамская НС	А.Г. Сбоев [305]	1850 текстов, написанных в рамках эксперимента и 41624 из социальных сетей (возрастные группы 18–23, 24–29, и старше 30 лет), пол – 2 класса. Средняя длина 300 символов. 4332 текста с искажением пола, 13632 с искажения возраста.	Частоты последовательностей символов 3-8-грамм, формирование признаков с учетом частей речи и эмоциональной составляющей	От 1000 текстов	Точность распознавания искажения возраста автора текста – 44% (3 класса), направления искажения – 80%.
ER-AE, DistilBERT	В. Панов и др. [150]	Тексты блогов на английском, испанском, португальском, итальянском, французском, немецком, японском, чешском и других языка. 750000 образцов	Сэмплирование на уровне слов, CNN	50 символов и более / 1000 текстов	Точность идентификации обфусцированных с помощью подхода текстов 50 авторов – менее 1%
SVM	S. Afroz и др. [7]	90 авторов, 450 текстов официально-делового или академического стиля, 450 текстов с искажением.	Количество символов, слов и предложений, число длинных слов, число букв в слове, количество коротких и длинных предложений, союзов, индекс удобочитаемости, количество модальных глаголов и числительных, количество упоминаний местоимения «я»	Не менее 500 слов / 5–10 текстов	2 класса – 97%
BERT, RoBERTa и ELECTRA	A. Hashemi и др. [97]	PAN-2017, 50000 текстов	Синтез подсказок	20–50 предложений / 500 текстов	10 авторов – 81%
DeBERTaV3, контрастивное обучение	Z. Ye и др. [225]			20-50 предложений / 500–1500 текстов	10 авторов – 75%
Forging the Forger	S. Corbara, A. Moreo [54]	TweepFake (40 авторов), EBG (10 авторов), Riddell-Juola (10 авторов), PAN11 (72 автора), Victoria (50 авторов)	Аугментация искусственно-сгенерированными данными	Не указано / 1 текст	TweepFake – 62%, EBG – 81%, Riddle-Juola – 61%, PAN11 – 56%, Victoria – 75%
Генеративные модели					
цепи Маркова, техника синонимизации с применением SIMP-таблиц	А.О. Исхакова [252]	3 группы по 3210 текстов: естественные (10,7 млн. символов), созданные с использованием цепей Маркова (14,1 млн	Эмоциональная окраска, соответствие жанру, стилю, соответствие текста времени	Не менее 1200 символов / 3210 текстов	2 класса – 93%

		символов), и созданные методом синонимизации (2,3 мл. символов)	его создания, логичность и связность		
Трансферное обучение, RuRoBERTa	С. Скрыльников [200]	215110 текстов, созданных людьми и 13 генеративными моделями	Неявное выделение признаков моделями	Не указано / 15365 текстов	14 классов – 65%
XGBoost	R. Shijaku и др. [195]	Эссе 126 людей, являющихся носителями английского языка и 126 текстов, созданных ChatGPT	TF-IDF, SHAP	250–300 слов / 126 текстов	2 класса – 93%
SVM, LR	A. Al Medawer [9]	500 научных статей на английском, созданных человеком, и столько же сгенерированных с использованием ИИ	Стилометрия, настроение, схожесть текста, читабельность и релевантность	Не указано / 500 текстов	2 класса – 99%
GPTZero, OpenAI Text Classifier, Writer.com's AI Content Detector, Copyleaks AI Content Detector и Giant Language model Test Room	С. Chaka и др. [45]	Тексты, сгенерированные ChatGPT, YouChat, и Chatsonic, 5000 текстов для каждой модели	Генеративные запросы, корректировка данных	1000–5000 символов / 110–2048 текстов	Не указано
RoBERTa	A. Uchendu и др. [205]	CTRL, GPT, GPT2, GROVER, XLM, XLNET, PPLM, FAIR, 1066 текстов созданы человеком и каждой из моделей	Метрики удобочитаемости	Не указано / 1 естественный текст, 8 искусственных	2 класса – 98%
SVM с отбором признаков, GRU+CNN	**А.С. Романов [66, 291, 186]**	**ВКонтакте – 2962 авторов, 325892текста, 6742 искусственных текста, 200 авторов, сгенерированы на основе естественных (GPT-3.5, GPT-4)**	***N*-граммы символов, распределение частей речи и знаков пунктуации, отбор генетическим алгоритмом**	**100-250 символов, / 50 образцов**	**Закрытая атрибуция: Только искусственные тексты 2 автора – 95%, 5 – 84%, 10 – 77%, 20 – 67%, 50 – 52% Внедрение искусственных текстов, как атака на метод авторства: 2 автора – 90%, 5 – 84%, 10 – 64%, 20 – 54%, 50 – 40%**

Атаки на методы идентификации автора исходного кода					
Запутывающие преобразования структур данных, графа потока управления, превентивные преобразования	Д.Н. Буинцев [237]	Программы различного типа, включая алгоритмы сортировки	Выделение базовых блоков исходного кода программы и построение графа потока управления, разрушение структуры графа потока управления, включая различные преобразования, введение массива переменных и их переприсваивание, введение «мертвого кода»	Не указано / около 16000 инструкций для исходной программы, увеличение до 20740 инструкций после запутывания	Вероятность взлома запутанной программы значительно ниже, чем исходной; показатель устойчивости повышается на 20–30%
Обфускация с использованием недостижимого кода	И.М. Кутузов [262]	150 программных модулей, разделенных на 3 класса по уровню запутанности: низкий, средний, высокий	Наличие недостижимого кода, число функций, вложенность циклов	100–200 строк / 50 файлов	3 класса – 85%
DL-CAIS, CSFS, WE-C-CNN и WE-S-CNN	M. Abuhamad и др. [5]	Датасет из 200 программистов с Google Code Jam, по 9 образцов на каждого программиста	Синтаксические, стилистические и структурные признаки, TFIDF-C и TFIDF-S	Среднее количество строк кода в образце: 68 / 1800 образцов в классе	Точность методов до атаки: от 84% до 98%, успешность нецелевых атак: >99%, целевых: 66–88%
Нечеткие АСД, экспертная оценка	L. Simko и др. [199]	Google Code Jam, 8661 программ, 3940 авторов, 3 набора данных (5, 20, 50 авторов)	Лексические, синтаксические, структурные признаки	Не указано / в среднем 43 файла на автора	2 класса (обфусцированный или без преобразований) – 80%
LSTM, RF	E. Quiring и др. [179]	C++, 1632 файла, 204 программиста	Лексические признаки, синтаксические признаки, признаки макета	74 строки / не указано	20 классов. Успешность атаки: 77% и 81% для имитации, 71% и 69% без шаблона
CNN, KNN	Z. Zafar и др. [128]	Google Code Jam, C++, Python и Java – обфусцированные образцы, 12408 авторов	Отображение входного потока символов в плотный вектор за счет размещения исходных кодов, созданных одним и тем же автором, близко друг к другу в векторном пространстве	15000–20000 символов, не указано	2 класса (обфусцированный или без преобразований) – 91%, Python – 85%, Java – 80%
CodeBERT	**А.С. Романов [258, 124, 64, 53, 261]**	**GitHub (13 языков, 102908 исходных кодов, 383 автора)**	**Структурные, лексические, нарушающие парадигмы программирования признаки**	**170 строк / 30 файлов**	**Средняя точность: Обфускация 83% Стандарты кодирования: 80% Искусственно-сгенерированный код: 94% Коммиты: 90%**

ПРИЛОЖЕНИЕ Б
ПСЕВДОКОДЫ АЛГОРИТМОВ

Алгоритм 1.1 – Трансферное обучение

```
{Задание переменных}
set dataset_path <- path_to_dataset;
set model_path <- path_to_bert_model;
set embeddings <- [];
set model <- classifier;
set data <- [];
set number_of_ngrams <- 3;
set learning_rate <-0.8;
set dimension <- 500;
set loss_function <- 'ova';
set max_number_of_allocated_memory_segments <- 2000000;
set cross_validation_results <- [];
{Процедура: Загрузка предварительно обученной модели BERT и замораживание слоев}
procedure Load_pretrained_bert_model(model_path):
begin
    BERT_Model = Load_Model(model_path);
    Freeze_Layers(BERT_Model);
    return BERT_Model;
end;
{Процедура: Загрузка и предобработка набора данных}
procedure Load_and_preprocess_dataset(dataset_path):
begin
    dataset = Load_Dataset(dataset_path);
    labels = dataset['labels'];
    texts = dataset['texts'];
    texts.remove_extra_spaces();
    texts.lower();
    return labels, texts;
end;
```

```
{Процедура: Получение векторов BERT для текстов}
procedure Get_bert_embeddings(texts):
begin
    bert_embeddings = embeddings;
    for each text in texts do
        tokenized_text = Tokenize Text with BERT Tokenizer;
        bert_embedding = Compute BERT Embeddings with BERT Model;
    end;
    return bert_embeddings;
end;
{Процедура: Сочетание метки и вектора в строку}
procedure Combine_label_and_embedding(label, embedding):
begin
    embedding_string = string(embedding);
    formatted_example = label + ‘ ‘ + embedding_string;
    return formatted_example;
end;
{Процедура: Сохранение данных в файл для GRU+CNN}
procedure Save_data_to_classifier_file(data, file_path):
begin
    file_path.open();
    for each formatted_example in data do
        file_path.write(formatted_example);
    end;
    file_path.close();
end;
{Процедура: Установка параметров обучения для GRU+CNN}
procedure Set_classifier_training_parameters(number_of_ngrams, learning_rate, dimension,
loss_function):
begin
    classifier(*params);
    return classifier;
end;
```

```
{Основной блок программы}
begin
  for each fold in 5_folds do
    training_data, validation_data = Split_dataset_into_fold(dataset_labels, dataset_bert_embeddings, fold);
    for each label, embedding in zip(training_data.labels, training_data.bert_embeddings) do
      formatted_example = Combine_label_and_embedding(label, embedding);
      data.append(formatted_example);
    end;
    classifier<- Set_training_parameters(**params);
    classifier.train(data);
    validation_accuracy = classifier.evaluate(validation_data);
    cross_validation_results.append(validation_accuracy);
  end;
end.
```

ПРИЛОЖЕНИЕ В
СВИДЕТЕЛЬСТВА О РЕГИСТРАЦИИ ПРОГРАММ ДЛЯ ЭВМ И БАЗ ДАННЫХ



ГОСУДАРСТВЕННАЯ АКАДЕМИЯ НАУК
РОССИЙСКАЯ АКАДЕМИЯ ОБРАЗОВАНИЯ
ИНСТИТУТ НАУЧНОЙ ИНФОРМАЦИИ И МОНИТОРИНГА
ОБЪЕДИНЕННЫЙ ФОНД ЭЛЕКТРОННЫХ РЕСУРСОВ "НАУКА И ОБРАЗОВАНИЕ"

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ЭЛЕКТРОННОГО РЕСУРСА

№ 15146

Настоящее свидетельство выдано на электронный ресурс, отвечающий требованиям новизны и приоритетности:

Программная система для идентификации автора письменной речи «Авторовед»

Дата регистрация: **24 декабря 2009 года**

Автор: **Романов А.С.**

Организация-разработчик: **ГОУ ВПО «Томский государственный университет систем управления и радиоэлектроники»**

Директор ИНИМ РАО,
чл.-корр. РАО, д.ю.н., проф. В.Е. Усанов

Руководитель ОФЭРНиО, почетный
работник науки и техники РФ А.И. Галкина

Дата выдачи **14.01.2010**

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2020666001

Программа для тонального анализа текста «Moodorrude»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Романов Александр Сергеевич (RU), Васильева Мария Игоревна (RU), Шилов Лев Сергеевич (RU), Куртукова Анна Владимировна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2020665103**

Дата поступления **26 ноября 2020 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***03 декабря 2020 г.***

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2021681140

Система для идентификации автора исходного кода программы "CoDEtective"

Правообладатели: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU)***

Заявка № **2021667210**
Дата поступления **26 октября 2021 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***17 декабря 2021 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 0x02A5CFBC00B1ACF59A40A2F08092E9A118
Владелец **Ивлиев Григорий Петрович**
Действителен с 15.01.2021 по 15.01.2035

Г.П. Ивлиев

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022616780

Чат-бот для определения возраста автора текстового сообщения «Age Detector»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Романов Алесандр Сергеевич (RU), Соболев Артем Александрович (RU), Федотова Анастасия Михайловна (RU), Куртукова Анна Владимировна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2022616248**
Дата поступления **12 апреля 2022 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***15 апреля 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b80077e14e40f0a94edbd24145d5c7
Владелец Зубов Юрий Сергеевич
Действителен с 2.03.2022 по 26.05.2023

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2022667584

Программный комплекс для получения информативных признаков автора текста «PyAuthorship»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2022667003**

Дата поступления **19 сентября 2022 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***22 сентября 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b80077e14e40f0a94edbd24145d5c7
Владелец Зубов Юрий Сергеевич
Действителен с 2.03.2022 по 26.05.2023

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2023685210

Программный комплекс для идентификации запрещенных законодательством РФ и деструктивных текстов «PyDestructiveDetector»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2023682992**

Дата поступления **01 ноября 2023 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***23 ноября 2023 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429b6a0fe3853164baf96f83b73b4aa7
Владелец Зубов Юрий Сергеевич
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2024685146

Программный комплекс для сбора данных о поле и гендере автора естественно-языкового текста

Правообладатель: ***Федеральное государственное автономное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2024683353**

Дата поступления **10 октября 2024 г.**

Дата государственной регистрации в Реестре программ для ЭВМ ***24 октября 2024 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 0692e7c1a6300bf154f240f670bca2026
Владелец **Зубов Юрий Сергеевич**
Действителен с 10.07.2024 по 03.10.2025

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2025615680

Программный комплекс для определения пола и гендера автора естественно-языкового текста «ProGender»

Правообладатель: ***Федеральное государственное автономное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2025613057**
Дата поступления **20 февраля 2025 г.**
Дата государственной регистрации
в Реестре программ для ЭВМ ***06 марта 2025 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 0692e7c1a6300bf54f240f670bca2026
Владелец **Зубов Юрий Сергеевич**
Действителен с 10.07.2024 по 03.10.2025

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации базы данных

№ 2022622299

«База данных естественно- и искусственно-языковых текстов для определения авторства»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Алесандр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2022622243**
Дата поступления **16 сентября 2022 г.**
Дата государственной регистрации
в Реестре баз данных ***21 сентября 2022 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 68b80077e14e40f0a94edbd24145d5c7
Владелец Зубов Юрий Сергеевич
Действителен с 2.03.2022 по 26.05.2023

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации базы данных

№ 2023623911

«База данных исходных и дизассемблированных кодов для идентификации авторства программ»

Правообладатель: ***Федеральное государственное бюджетное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2023623657**
Дата поступления **01 ноября 2023 г.**
Дата государственной регистрации
в Реестре баз данных ***13 ноября 2023 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 429b6a0fe3853164baf96f83b73b4aa7
Владелец Зубов Юрий Сергеевич
Действителен с 10.05.2023 по 02.08.2024

Ю.С. Зубов

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации базы данных

№ 2024624287

«База данных естественных текстов для идентификации пола и гендера автора текста»

Правообладатель: ***Федеральное государственное автономное образовательное учреждение высшего образования «Томский государственный университет систем управления и радиоэлектроники» (RU)***

Авторы: ***Куртукова Анна Владимировна (RU), Романов Александр Сергеевич (RU), Федотова Анастасия Михайловна (RU), Шелупанов Александр Александрович (RU)***

Заявка № **2024624158**
Дата поступления **05 октября 2024 г.**
Дата государственной регистрации
в Реестре баз данных ***14 октября 2024 г.***

Руководитель Федеральной службы по интеллектуальной собственности

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 0692e7c1a6300bf54f240f670bca2026
Владелец **Зубов Юрий Сергеевич**
Действителен с 10.07.2024 по 03.10.2025

Ю.С. Зубов

ПРИЛОЖЕНИЕ Г
АКТЫ ВНЕДРЕНИЯ И ИСПОЛЬЗОВАНИЯ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ

ТУСУР

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»

ОКПО 02069326, ОГРН 1027000867068,
ИНН 7021000043, КПП 701701001
пр. Ленина, 40, г. Томск, 634050

тел: (382 2) 510-530
факс: (382 2) 513-262, 526-365
e-mail: office@tusur.ru
https:// www.tusur.ru

«Утверждаю»
Проректор по учебной работе Томского государственного университета систем управления и радиоэлектроники
_____ П.В. Сенченко
«26» сентября 2024 г.

АКТ
О внедрении результатов диссертационной работы Романова Александра Сергеевича в учебный процесс

Комиссия в составе:
Председателя:
Давыдовой Е.М., канд. техн. наук, доцента, декана факультета безопасности ТУСУРа
членов комиссии:
Конева А.А., канд. техн. наук, доцента каф. КИБЭВС ТУСУРа
Костюченко Е.Ю., канд. техн. наук, доцента каф. КИБЭВС ТУСУРа
Новохрестовой Д.И., канд. техн. наук, доцента каф. КИБЭВС ТУСУРа
составила настоящий Акт о нижеследующем.

Результаты диссертационной работы Романова А.С. используются в учебном процессе на факультете безопасности ТУСУРа при изучении дисциплин «Языки программирования» и «Основы построения систем искусственного интеллекта и машинного обучения» для подготовки специалистов, обучающихся по специальностям 10.05.02 «Информационная безопасность телекоммуникационных систем», 10.05.04 «Информационно-аналитические системы безопасности», 10.05.03 «Информационная безопасность автоматизированных систем», бакалавров по направлению 10.03.01 «Информационная безопасность», магистров по направлению

09.04.04 «Программная инженерия».

Результаты диссертационной работы также используются при выполнении дипломных, курсовых, научно-исследовательских и групповых проектных работ студентами факультета безопасности ТУСУРа.

На основе положений диссертационной работы Романова А.С. разработан комплекс практических и лабораторных работ, включающий методическое и программное обеспечение для реализации подходов оценки информативности характеристик текста в задачах кибербезопасности, связанных с анализом текста. В ходе выполнения работ студенты имеют возможность проводить исследование текстов и их характеристик для решения задач идентификации автора, возраста, пола, настроения автора. Методы анализа включают как классические методы машинного обучения и нейросетевые подходы, так и современные глубокие архитектуры нейронных сетей.

Методика проверки однородности текста и поиска заимствований используется на факультете безопасности для оценки работ студентов. Использование методики и программных модулей на ее основе позволило сократить время преподавателя на проверку студенческий работ на 20%.

Председатель комиссии:

Декан ФБ ТУСУРа, канд. техн. наук _____ Давыдова Е.М.

Члены комиссии:

Доцент каф. КИБЭВС ТУСУРа, канд. техн. наук. _____ Конев А.А.

Доцент каф. КИБЭВС ТУСУРа, канд. техн. наук. _____ Костюченко Е.Ю.

Доцент каф. КИБЭВС ТУСУРа, канд. техн. наук. _____ Новохрестова Д.И.

Утверждаю

Первый заместитель командира

войсковой части 51952

полковник А.Ю.Капитанов

«11» мая 2010 г.

АКТ

о внедрении результатов диссертационной работы Романова Александра Сергеевича

Комиссия в составе: председателя – заместителя командира части полковника Новикова Ю.А. и членов – заместителя начальника отдела подполковника Косолапова С.А. и ведущего сотрудника отдела майора Лопатина Е.А. составили настоящий акт в том, что в войсковой части 51952 внедрены методики идентификации автора неизвестного текста и элементы программной системы «Автороведъ», разработанные Романовым А.С.

В ходе оценки эффективности полученных методик проводилось сравнение точности идентификации автора текста длиной 100 символов в случае двух возможных претендентов, а также необходимых для этого временных затрат. Для обучения использовалось по 50 электронных сообщений, тестирование проводилось на 25 сообщениях каждого автора. Всего было проведено 830 экспериментов. В результате исследований были получены следующие показатели:

– точность идентификации с помощью многослойного перцептрона составила 0,62±0,01, среднее время обучения одной сети 503,1 с.;

– точность идентификации с помощью сетей каскадных корреляций составила 0,70±0,17, среднее время обучения одной сети 206,2 с.;

– точность идентификации с помощью машины опорных векторов составила 0,69±0,12, среднее время обучения одной модели 0,6 с.;

Полученные результаты позволяют сделать вывод о возможности идентификации авторства коротких текстов с помощью разработанных Романовым А.С. методик и программной системы. Наиболее перспективной представляется методика, использующая машину опорных векторов.

Председатель комиссии
Заместитель командира части – начальник отдела
кандидат технических наук
полковник Ю.А.Новиков

Члены комиссии
Заместитель начальника отдела
подполковник С.А.Косолапов

Ведущий сотрудник отдела
кандидат технических наук
майор Е.А.Лопатин

НАЦИОНАЛЬНЫЙ ИННОВАЦИОННЫЙ ЦЕНТР
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

117246, г. Москва,
Научный проезд, д.6.
Тел.: +7 (495) 204-20-86
info@centin.ru
www.centin.ru

УТВЕРЖДАЮ
Технический директор Акционерного общества «Национальный Инновационный Центр», кандидат технических наук

/ Зыков В.Д.
«__» августа 2024 г.

АКТ
о внедрении результатов диссертационной работы
на соискание ученой степени доктора технических наук
Романова Александра Сергеевича

Комиссия в составе:

Председатель:		
	Зыков В.Д.	Технический директор Акционерного общества «Национальный Инновационный Центр», кандидат технических наук
Члены комиссии:		
	Козинов С.Ю.	Руководитель проектов Акционерного общества «Национальный Инновационный Центр»
	Гайдуков Л.П.	Эксперт отдела обеспечения информационной безопасности автоматизированных систем Акционерного общества «Национальный Инновационный Центр»
	Зуева А.А.	Эксперт отдела обеспечения информационной безопасности автоматизированных систем Акционерного общества «Национальный Инновационный Центр»

составила настоящий Акт о том, что в деятельность Акционерного общества «Национальный Инновационный Центр» (АО «НИЦ») по проектированию, разработке, внедрению и эксплуатации систем защиты информации были внедрены

результаты диссертационной работы Романова А.С. на соискание ученой степени доктора технических наук.

Комиссия установила, что:

1) при разработке средств многофакторной аутентификации использовались Методики верификации и продленной аутентификации пользователя по тексту. Это позволило повысить точность существующей системы аутентификации на 8% без применения системы одноразовых паролей;

2) при разработке системы защиты персональных данных использовались Методики идентификации автора текста и определения пола и возраста. Это позволило сократить временные затраты на 15-20% при обработке больших массивов текстовых данных;

3) при разработке систем анализа сетевого трафика использовалась Методика определения текстов деструктивной и экстремистской направленности. Это позволило повысить точность фильтрации нежелательного трафика, содержащего, в том числе, экстремистские материалы на 5%.

Таким образом, разработанная Романовым А.С. методология позволяет эффективнее реализовывать проекты, связанные с импортозамещением специализированного программного в области защиты информации.

Председатель комиссии:

Зыков В.Д. _____

«1» августа 2024 г.

Члены комиссии:

Козинов С.Ю. _____

«1» августа 2024 г.

Гайдуков Л.П. _____

«1» августа 2024 г.

Зуева А.А. _____

«1» августа 2024 г.

sibedge

ООО «Сибэдж», LLC «SibEDGE»
634006, Томск, ул. Пушкина, 75
тел: +7 (38-22) 701-841

ОГРН 1127017015179
ИНН/КПП 7017307138 / 701701001
РЕГ.НОМЕР 7000037783

«Утверждаю»

Технический директор ООО «Сибэдж»

_____ Иванов Н.А.

«07» февраля 2024 г.

Комиссия в составе: председателя – руководителя отдела интеграции бизнес-процессов ООО «Сибэдж» Алеутдинова О.В. и членов – старшего бизнес-аналитика ООО «Сибэдж» Дробинской М.А., ведущего инженера-программиста ООО «Сибэдж» Лисунова А.А., руководителя проектов ООО «Сибэдж» Абдурахимовой З.

составила настоящий Акт о том, что в ООО «Сибэдж» были апробированы, частично реализованы и внедрены результаты диссертационной работы Романова А.С. на соискание ученой степени доктора технических наук. Комиссия установила, что:

1) методика определения текстов деструктивной направленности, алгоритмы и программное обеспечение «PyDestructiveDetector», разработанные Романовым А.С. применяются в разрабатываемых ООО «Сибэдж» программных продуктах, предназначенных для выявления и фильтрации негативных комментариев в социальных сетях и мессенджерах, определения их авторов, реакций пользователей на такие сообщения и оперативного реагирования на угрозы безопасности, возникающих вследствие этого;

2) использование алгоритмов и программных модулей, разработанных Романовым А.С., позволило повысить точность определения негативных текстовых материалов на 15% (до 95%), определять их автора с точностью 84%;

3) вместе с тем временные затраты на обработку больших массивов текстовых данных сократились в 8 раз, что позволило реализовать механизмы модерации и фильтрации текстового трафика в режиме реального времени.

Председатель комиссии:

Руководитель отдела интеграции бизнес-процессов ООО «Сибэдж» — Алеутдинов О.В.

Члены комиссии:

старший бизнес-аналитик ООО «Сибэдж» — Дробинская М.А.

ведущий инженер-программист ООО «Сибэдж» — Лисунов А.А.

руководитель проектов ООО «Сибэдж» — Абдурахимова З.

ООО «НТР ТОМСК»
634055, г. Томск, проспект Развития, дом 3, офис 607.
ИНН 7017118839 КПП 701701001 ОГРН 1057000156927
+7 3822 488 532 || info@ntrl.ai || ntr.ai

«Утверждаю»
Директор ООО «НТР Томск»
_____ Михайловский Н.Э.
«28» 08 2024 г.

Комиссия в составе: председателя – ведущего руководителя проектов ООО «НТР Томск» Нургалиева Олега Тахировича и членов – директора проектного офиса ООО «НТР Томск» Менщикова Сергея Николаевича и руководителя проектов ООО «НТР Томск» Барышева Ильи Игоревича

составила настоящий Акт о том, что ООО «НТР Томск» были апробированы, частично реализованы и внедрены результаты диссертационной работы Романова А.С. на соискание ученой степени доктора технических наук. Комиссия установила, что:

1) алгоритмы и методики идентификации пола и возраста автора текста, разработанные Романовым А.С. применяются в разрабатываемых ООО «НТР Томск» рекомендательных системах на основе интеллектуального анализа текста для предоставления персонализированных предложений пользователям;

2) точность предоставления рекомендаций пользователям, относящимся к разным возрастным категориям, за счет внедренных положений диссертации выросла на 5%;

3) точность предоставления рекомендаций пользователям, относящимся к разному полу, за счет внедренных положений диссертации выросла на 6%;

4) использование алгоритмов и программных модулей, разработанных Романовым А.С., позволили сократить временные затраты на разметку текстовых данных для задач по обработке естественного языка на 11% и учесть атрибуты авторов текста.

5) использование Методики идентификации автора исходного кода программы позволяет минимизировать репутационные риски компании.

Председатель комиссии:
Ведущий руководитель проектов ООО «НТР Томск», к.т.н. _____ Нургалиев О.Т.

Члены комиссии:
директор проектного офиса ООО «НТР Томск» _____ Менщиков С.Н.
руководитель проектов ООО «НТР Томск» _____ Барышев И.И.

Общество с ограниченной ответственностью
«Системы информационной безопасности»
ИНН 5405414231 КПП 540501001 ОКПО 66225083
Р/счет: 40702810120240007723 Филиал «Центральный» Банка ВТБ (ПАО) в г. Москве
К/счет: 30101810145250000411 БИК: 044525411
630008, г. Новосибирск, ул. Бориса Богаткова, д. 67, офис 8
тел.: +7 (383) 201-85-75, info@sib-nsk.net

«Утверждаю»
Директор ООО «СИБ»
А.А. Помешкин
«10» октября 2024 г.

АКТ

О внедрении результатов диссертационной работы

на соискание ученой степени доктора технических наук

Романова Александра Сергеевича

Комиссия в составе: председателя – В.В. Тюменцева, заместителя руководителя центра ГосСОПКА ООО «СИБ» и членов – А.А. Вяткина, начальника отдела внедрения и сопровождения информационных систем ООО «СИБ», А.Г. Марковина, главного специалиста отдела по внедрению и эксплуатации программно-технических средств ООО «СИБ», А.А. Балабанова, специалиста отдела внедрения и сопровождения информационных систем ООО «СИБ», составили настоящий акт о том, что результаты диссертационной работы А.С. Романова были внедрены в ООО «СИБ».

Методология идентификации автора текстовой информации для решения задач кибербезопасности, включая методики определения автора естественного текста, его возраста, пола и гендера, а также обнаружения текстов деструктивной и экстремистской тематики используется для разработки защищенных информационных систем, в том числе реализации механизма продленной аутентификации пользователей по стилю текста, а также системы фильтрации сетевого трафика.

В ходе испытаний был проведен анализ текстов 2962 пользователей с 50 образцами для каждого. Длина предоставленных образцов составила от 50 до 250 символов, все тексты написаны на русском языке. Из них 1500 пользователей были младше 18 лет, 960 – в возрасте от 21 до 27 лет и 502 – старше 30 лет. 50% текстов написаны мужчинами, 50% – женщинами.

В процессе оценки эффективности методики определения автора текста выполнялось сравнение точности для различных выборок на основе набора данных.

В результате были получены следующие показатели:

1. Средняя точность идентификации среди 2 авторов – 96%.
2. Средняя точность идентификации среди 5 авторов – 94%.
3. Средняя точность идентификации среди 10 авторов – 78%.
4. Средняя точность идентификации среди 20 авторов – 74%.
5. Средняя точность идентификации среди 50 авторов – 56%.

Применение методик позволило реализовать механизм продленной аутентификации пользователей, снизить затраты и временными ресурсы компании на разработку датасетов и инструментов, требуемых для верификации пользователей в два раза, обеспечивая высокую точность распознавания авторов – до 96%.

Также была проведена классификация аудитории на детскую и взрослую возрастные группы, и определение пола. Точность определения возраста составила 85%, пола – 90%. Результаты показали, что использование методики определения возраста автора текста позволяет учесть текстовую составляющую трафика и повысить общую эффективность системы безопасного цифрового взаимодействия на 13% для снижения нежелательного трафика для детской аудитории.

Разработанной А.С. Романовым методикой обнаружения деструктивного контента и соответствующим комплексом программ проанализировано два набора данных, содержащих комментарии обычных пользователей и потенциально экстремистские и дискредитирующие сообщения. В первом случае набор данных включал 40000 образцов (по 20000 в каждом классе), во втором случае 30000 образцов (по 15000 в классе). Было выявлено 18800 образцов, содержащих экстремистский контент (точность 94%) и 13504 текстов, содержащих деструктивный контент (точность 90%).

С использованием методики поиска пропаганды нетрадиционных ценностей было проанализировано 90000 образцов текстов, созданных мужчинами, женщинами и представителями ЛГБТ-сообщества (30000 образцов в каждом классе). На основе полученных результатов установлено, что 27312 текстов являются созданными представителями ЛГБТ (точность 91%).

Техническое преимущество разработанных А.С. Романовым инструментов стоит в автоматизации процессов сканирования и анализа данных, повышении скорости их обработки до 80 раз по сравнению с ручной модерацией, что обеспечивает не только экономию времени, но и повышает общую эффективность мониторинга. Внедрение методик позволило обеспечить своевременное реагирование на текстовый контент, созданный запрещенными на территории Российской Федерации организациями и их последователями или содержащий пропаганду нетрадиционных ценностей, обнаружение распространителей такого контента и установление их возраста, пола и гендера.

Председатель комиссии:

В.В. Тюменцев «10» октября 2024 г.

Члены комиссии:

А.А. Вяткин «10» октября 2024 г.

А.Г. Марковин «10» октября 2024 г.

А.А. Балабанов «10» октября 2024 г.

МОСКОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В.ЛОМОНОСОВА
(МГУ)
ЭКОНОМИЧЕСКИЙ ФАКУЛЬТЕТ

Ленинские горы, Москва, д.1, стр.46, 119234
Телефон: (495) 939-33-76, Факс: (495) 939-08-77

04.12.24 № 822-24/114-03/4

АКТ

О внедрении результатов диссертационной работы

на соискание ученой степени доктора технических наук

Романова Александра Сергеевича

Экономический факультет МГУ имени М.В. Ломоносова подтверждает, что результаты диссертационной работы Романова А.С. были апробированы и внедрены на кафедре народонаселения экономического факультета МГУ имени М.В. Ломоносова для определения демографических характеристик (пола и возраста) комментаторов социальных сетей в рамках работы над темой «Демографические детерминанты оценки качества медицинских услуг и отказа от табакокурения: анализ мнений россиян на основе применения нейросетей и генеративного искусственного интеллекта».

С помощью разработанного Романовым А.С. программного комплекса были размечены данные по отказу от табака (165480 записей) и отзывы пациентов клиник (66364 записи) с целью получить альтернативную оценку демографических характеристик комментаторов социальных сетей. Подход Романова А.С. к автоматизации демографических характеристик авторов текстов в социальных сетях позволяет увеличить объемы классифицируемых данных и скорость получения результата. Точность определения пола составила 85-89% в зависимости от датасета, точность определения возраста составила 71-86% в зависимости от возрастной категории и датасета. Особое преимущество данного подхода в классификации возраста – определение подростковой группы комментаторов.

Внедрение результатов диссертационной работы Романова А.С. позволяет принимать во внимание возраст и пол пользователей социальных сетей при решении прикладных задач демографии, связанных с улучшением политики в области народонаселения (семейной, демографической, миграционной политики, политики по совершенствованию системы здравоохранения).

Руководитель НИР, заведующая кафедрой народонаселения экономического факультета МГУ имени М.В. Ломоносова, д-р. экон. наук, проф. И.Е.Калабихина

Заместитель декана экономического факультета МГУ имени М.В. Ломоносова по научной работе, канд.экон.наук А.А. Курдин

Подпись Калабихиной И.Е. Курдина А.А.
удостоверяю
Зав. канцелярией экономического
факультета МГУ

«Утверждаю»
Врио начальника ОБК
УМВД России по Томской области
полковник полиции
_____С.А.Изосимов
«28» 11 2024г.

АКТ
О внедрении результатов диссертационной работы
на соискание ученой степени доктора технических наук
Романова Александра Сергеевича

Комиссия в составе: председателя – старшего оперуполномоченного ОБК УМВД России по Томской области старший лейтенант полиции Аксенов В.Д. и члены – старший оперуполномоченный ОБК УМВД России по Томской области младший лейтенант полиции Малетина Е.А., оперуполномоченный ОБК УМВД России по Томской области старший лейтенант полиции Догот Р.В.составили настоящий акт о том, что результаты диссертационной работы Романова А.С. были внедрены в Отдел по борьбе с противоправным использованием информационно-коммуникационных технологий УМВД России по Томской области в виде методики поиска деструктивного контента и пропаганды нетрадиционных ценностей и программной системы PyDestructiveDetector.

Разработанной Романовым А.С. методикой и соответствующим комплексом программ проанализировано два набора данных, содержащих комментарии обычных пользователей и потенциально экстремистские и дискредитирующие сообщения. В первом случае набор данных включал 43518 образцов, во втором случае – 31376 образцов. Было выявлено 19108 образцов, содержащих экстремистский контент – точность 94%, и 13504 текстов, содержащих деструктивный контент – точность 90%. Результаты ручной экспертной проверки совпали с результатами, полученными на основе методики на 95%. Установлено, что использование методики поиска деструктивного контента позволило сократить временные затраты на анализ текста в 15 раз, по сравнению с ручной проверкой.

С использованием методики поиска пропаганды нетрадиционных ценностей было проанализировано еще 92243 образцов текстов, созданных мужчинами, женщинами и представителями ЛГБТ-сообщества. На основе полученных результатов установлено, что 27312 текстов являются созданными представителями ЛГБТ – точность 91%.

Техническое преимущество разработанной системы стоит в ее способности автоматизировать процессы сканирования и анализа данных, повысить скорость их обработки в 15 раз по сравнению с ручной проверкой. Применение методики позволило повысить эффективность отдела на 7%.

Председатель комиссии:

Старший оперуполномоченный ОБК
УМВД России по Томской области
старший лейтенант полиции В.Д. Аксенов

Члены комиссии:

Старший оперуполномоченный ОБК
УМВД России по Томской области
младший лейтенант полиции Е.А. Малетина

Оперуполномоченный ОБК
УМВД России по Томской области
старший лейтенант полиции Р.В.Догот

TOMSK POLYTECHNIC UNIVERSITY ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

«Утверждаю»
Директор инженерной школы информационных технологий и робототехники ТПУ
А.С. Фадеев
«19» декабря 2024 г.

ИШИТР

АКТ
О внедрении результатов диссертационной работы
на соискание ученой степени доктора технических наук
Романова Александра Сергеевича

Комиссия в составе:

- председателя – Шерстнёва В.С., канд. техн. наук, руководителя Отделения информационных технологий (ОИТ) ИШИТР ТПУ, доцента ОИТ ИШИТР ТПУ,

и членов:

- Губина Е.И., канд. физ.-мат. наук, руководителя ООП «Технологии и анализ больших данных», доцента ОИТ ИШИТР ТПУ,
- Савельева А.О., канд. техн. наук, зав. лаб. ОИТ ИШИТР ТПУ, доцента ОИТ ИШИТР ТПУ,
- Саврасова Ф.В., канд. техн. наук, доцента ОИТ ИШИТР ТПУ,

составила настоящий акт о том, что результаты диссертационной работы Романова А.С. были внедрены в лаборатории ИШИТР ТПУ.

Методики идентификации автора текста, программы PyAuthorship и CoDEtective, применяются для детектирования ботов и генеративных моделей семейства GPT при создании текста, выявления попыток имитации авторского стиля письма, а также обнаружения искусственно созданных работ, в том числе текстовых объектов авторской собственности, размещаемых в сети Интернет.

При апробации методики использовались комментарии пользователей социальных сетей, образцы студенческих работ и фрагменты исходного кода (всего 9500 текстов от 190 человек) и созданные генеративными моделями семейств GPT-3.5, GPT-4 на их основе тексты (1500 искусственных текстов). В ходе испытаний следовало выявить тексты, сгенерированные GPT-моделями. Также оценка методики проводилась в условиях внедрения генеративных образцов, созданных на основе стилей авторов, чьи тексты используются в ходе испытания для имитации условий атаки на метод. Проверка была проведена для выборок из 2-50 авторов.

Эксперименты показали, что применение методик и программ, разработанных Романовым А.С., позволяет решать поставленные задачи, учитывать тексты, созданные генеративными алгоритмами, снизить вовлеченность экспертов и сократить их временные затраты при анализе текстовых документов в отдельных случаях в 6 раз. Это делает результаты диссертационной работы Романова А.С. особенно полезными в образовательной деятельности для оценки качества студенческих работ и в области защиты авторских прав.

Председатель комиссии:
Руководитель ОИТ ИШИТР ТПУ, канд. техн. наук, доцент ОИТ ИШИТР ТПУ Шерстнёв В.С.

Члены комиссии:
Руководитель ООП «Технологии и анализ больших данных», канд. физ.-мат. наук, доцент ОИТ ИШИТР ТПУ Губин Е.И.
Зав. лаб. ОИТ ИШИТР ТПУ, канд. техн. наук, доцент ОИТ ИШИТР ТПУ Савельев А.О.
Канд. техн. наук, доцент ОИТ ИШИТР ТПУ Саврасов Ф.В.