ОГЛАВЛЕНИЕ

ОТ СОСТАВИТЕЛЕЙ

Настоящий тематический выпуск журнала «Электронные библиотеки» включает статьи, подготовленные на основе докладов, представленных на Всероссийской конференции с международным участием «Актуальные проблемы семантического анализа данных», которая прошла 16–19 февраля 2026 г. в г. Коломна. Конференция по такой тематике была проведена впервые и была посвящена 80-летию со дня рождения Владимира Алексеевича Серебрякова (1946–2024).

Ключевые направления конференции:

- интеллектуальный анализ данных;
- извлечение знаний, анализ научных данных;
- онтологии, графы знаний и управление знаниями;
- нейросимволические и LLM-ориентированные методы анализа данных;
- информационный поиск и анализ текстов;
- цифровые библиотеки, метаданные и научная коммуникация;
- семантический поиск, оценка качества, безопасность;
- интеллектуальный анализ данных в задачах информационной безопасности.

Основными целями проведенной конференции были объединение специалистов, исследователей и студентов для обсуждения современных актуальных задач семантического анализа данных, обмена результатами и опытом, а также содействие междисциплинарному научному диалогу. Конференция организована сотрудниками Федерального исследовательского центра «Информатика и управление» Российской академии наук и поддержана Российской ассоциацией искусственного интеллекта.

Редакторы-составители: К. В. Воронцов, Е. К. Липачёв, Н. П. Тучкова

УДК 004.8

ОРКЕСТРАЦИЯ МЕТОДОВ АНАЛИЗА НАУЧНЫХ ДАННЫХ В ПРОЦЕССАХ РЕЦЕНЗИРОВАНИЯ

О. М. Атаева[1] [0000-0003-0367-5575], Н. П. Тучкова[2] [0000-0001-5357-9640]

[1, 2]Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия

[1]oataeva@frccsc.ru, [2]ntuchkova@frccsc.ru

Аннотация

Исследована проблема сочетания методов в задаче семантического анализа научных данных и публикаций при рецензировании. На разных этапах обработки данных в системе SciLibRu использованы различные методы, построена многоуровневая онтология, наполнен граф знаний, что приводит к формированию новой структуры данных, отличной от исходной. Каждый метод по отдельности приобретает свое назначение в такой системе, при этом в совокупности их сочетание приводит к возникновению новых свойств, которые стали предметом настоящих исследований. Приведен пример автоматического агента рецензирования с объяснимым результатом.

Ключевые слова: *оркестрация методов, семантический анализ, онтология предметной области, граф знаний, большие языковые модели, системы, категории, динамические структуры.*

ВВЕДЕНИЕ

Одним из парадоксов цифровизации стало то, что искусственному интеллекту (ИИ) «отдаются» творческие задачи, которые выполняют большие языковые модели (БЯМ), а человеку остаются рутинные задачи, такие как разметить текст, вычитать рукопись, исправить подписи к рисункам в статьях и т. д. Ученые продолжают активно учить БЯМ и восхищаются их способностям «рассуждать», но в то же время сетуют, что скоро труд исследователя обесценится и сведется к «правильной постановке вопроса» для БЯМ.

Проблема использования накопленной цифровой информации для сопровождения научных исследований по-прежнему остается актуальной. Еще один из парадоксов состоит в том, что методов обработки данных, которые как раз создавались для снятия рутинной нагрузки, огромное количество, но они не демонстрируют системного подхода.

Продукты ИИ, такие как система Prism OpenAI [1], российская разработка DoTrace [2] и др., ускоряют написание научных текстов, что вызывает вопрос о том, можно ли доверять этим результатам, считать ли их научными и как их рецензировать. Объем публикаций растет, инструменты множатся и при этом нарастает фрагментация: каждый метод решает свою задачу изолированно. Большое количество работы с данными не приводит к упорядочиванию творческого процесса исследователя и рецензента, а способствует росту хаоса в этой сфере (вспомним второй закон термодинамики и рост энтропии [3]).

Естественным становится желание вернуться к первоначальной задаче автоматизации работы с данными, использованию семантического анализа для извлечения знаний, применению всего комплекса методов для сопровождения научных исследований и рецензирования. Необходим переход к системной архитектуре, где граф знаний (ГЗ) может играть роль координационного ядра. Оркестрация методов служит такой цели и используется применительно к различным системам данных и управления [4], она связана с понятием самоорганизации и моделирования интеллектуальных агентов.

Мы будем обращаться к методам, предназначенным для представления знаний научной предметной области, для формального описания агентов и управления оркестрацией в интеллектуальной системе поддержки научных исследований. Для организации процессов предлагается использовать модель многоуровневой онтологии, которая организована по принципу разделения ответственности и включает пять взаимосвязанных уровней. Нижние уровни описывают объекты предметной области и информационные ресурсы. Средний уровень является ядром оркестрации, он представляет методы обработки как полноправные онтологические объекты с явно декларированными предусловиями и постусловиями. Верхние уровни описывают динамику поведения агентов с их возможностями и сам процесс оркестрации. Межуровневые отображения позволяют «рассуждателю» (системе логических рассуждений, reasoning

engine/system) выводить применимые методы автоматически, без жестко закодированной логики маршрутизации. Активизация различных методов при инициализации запросов с помощью БЯМ на естественном профессиональном языке задает динамику взаимодействия и самоорганизации модулей системы на основе формальных описаний, заложенных в архитектуре.

Под оркестрацией понимается как организация процессов в информационной системе на основе специальной структуры, где на разных уровнях управления данными применяются различные методы и/или их сочетание.

На примере задачи рецензирования научной публикации предложено описание цифрового агента формирования шаблона для эксперта.

1. БЛИЗКИЕ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

В современных информационных системах развиваются идеи интеграции на основе семантической оркестрации, эмерджентности, самоорганизации и динамических структур. Подходы самоорганизации порождают новые архитектуры, основанные на моделировании динамики интеллектуальных агентов. Ключевое направление составляют идея эволюции мультиагентных систем и переход от статических структур к динамическим, основанным на привлечении человека как эксперта в системах принятия решений.

В исследовании [5] предложена идея перехода от мультиагентной системы к интегрированной динамической адаптивной системе принятия решений, где интеллектуальные агенты функционируют совместно с экспертной оценкой человека. Авторы предлагают ввести социоэкономические метрики оценки эффективности системы оркестрации.

На фоне перечисленных тенденций изменяется сама организация взаимодействия программных модулей. Возникают системы, где управление ГЗ и онтологией инициализируется с помощью БЯМ. В работе [6] предложено несколько методологий построения ГЗ с применением БЯМ которые, в свою очередь, используют различные подходы на разных этапах процесса построения ГЗ. Дан обзор методов построения онтологий и ГЗ на основе динамических структур. Рассмотрены нисходящие и восходящие парадигмы моделирования онтологий, где БЯМ вносят изменения в формальные описания и коррекции при генерации он-

тологий. Нисходящая парадигма делает акцент на логике проектирования, используя БЯМ для преобразования входных данных на естественном языке в формальные онтологии таких стандартов, как OWL. Авторы работы [6] утверждают, что их подход значительно повышает согласованность и сложность взаимодействия в системе, используя метакогнитивные подсказки для обеспечения саморефлексии и структурной коррекции во время генерации онтологий.

Привлечение БЯМ к оркестрации методов не только на уровне взаимодействия моделей, но и на уровне построения онтологий приводит к тиражированию ошибок, которые выдают галлюцинации на реальных данных. Такие ошибки, которые получили название “bias” (предвзятость) [7], образуются при переводе и миграции понятий и терминов. Далее, при настройке на определенную предметную область происходят «смещение» или «предвзятость» понятий, которые невозможно устранить, поскольку они заложены в онтологии. Требуется переобучение БЯМ или изменение онтологии, т. е. применение дополнительных методов корректировки семантики информационных систем. В работе [7] и многочисленных источниках, на которые ссылаются авторы этого исследования, предложено на практике привлекать экспертов и/или дополнять знания тематическими данными, чтобы определять фрагменты “bias” и устранять их для повышения достоверности при поиске.

Вопросам семантики оркестрации посвящено исследование [8], где предложен инструментарий для формального описания условий корректности семантической оркестрации методов анализа в различных предметных областях. В [8] введено понятие логических условий (Validity Constraints, VC), которые должны выполняться в определенные моменты рабочего процесса анализа данных. Условия VC задают как инварианты корректности для промежуточных состояний конвейера обработки данных и как инструменты для обнаружения нарушений. В процессе оркестрации должно выполняться формальное задание ограничений и их проверка.

Заметим, что традиционная роль онтологии сводится к описанию предметной области. В условиях агентных интеллектуальных систем, где необходимо динамически выбирать варианты обработки запроса и компоновать разнородные методы семантического анализа от символьного SPARQL-поиска и векторного поиска по эмбеддингам до БЯМ-генерации и формальной верификации,

возникает потребность в дополнительном уровне – *онтологическом описании самих процессов и методов агента*.

Настоящая статья посвящена разработке многоуровневой онтологии, которая служит единой архитектурной основой для представления научных знаний предметной области и формального описания методов агента и управления их оркестрацией. Онтология в данном подходе первична по отношению к ГЗ. Граф представляет собой материализацию онтологии на конкретных данных предметной области, а оркестратор – функциональный компонент навигации по графу с помощью БЯМ и посредством онтологического рассуждения.

2. ОНТОЛОГИЧЕСКИЙ ПОДХОД К СЕМАНТИЧЕСКИМ НАУЧНЫМ БИБЛИОТЕКАМ

2.1 Источники, особенности и представление научных данных

Научные данные могут быть востребованы во многих задачах: это поиск, сравнение, структурирование, формализация и другие. Каждая задача требует своего метода, а данные могут быть различной природы: текст, формулы, изображения, графовые структуры и др. Оркестрация позволяет работать с этими и другими объектами информационной системы в единой структуре. Настоящее исследование посвящено оркестрации как сочетанию методов обработки, подобранных для конкретных задач и конкретных данных.

Чтобы понять масштаб поставленной задачи, необходимо оценить природу современных научных данных. Они больше не ограничиваются традиционными публикациями. Мы имеем классические статьи и монографии, энциклопедии и классификаторы, формальные библиотеки доказательств. Но все чаще рождаются материалы, созданные с участием ИИ: автоаннотации, синтетические обзоры, автоматически сгенерированные формализации.

Научные данные обладают рядом специфических свойств, которые делают их обработку особенно сложной. Перечислим эти свойства:

– узкоспециализированная терминология и сложные иерархии понятий. Одно и то же понятие может иметь разные интерпретации в различных контекстах;

– знание распределено между формальными и неформальными представлениями: доказательство может существовать одновременно в строгом формальном виде и в текстовом объяснении;

– научные области динамичны, появляются новые концепты, изменяются связи, возникают междисциплинарные направления;

– корпус знаний многоязычен и неоднороден по стилю и структуре. Эти особенности делают очевидным то, что простое сопоставление слов или векторная близость не обеспечивают полноценного понимания структуры знания;

– центральной структурой хранения и навигации по научным знаниям в современных интеллектуальных системах служит ГЗ, обеспечивающий структурированное представление сущностей предметной области и их отношений.

Для работы с классическими источниками была разработана семантическая библиотека LibMeta [9], где в основу модели данных была заложена онтология. Работа с искусственными источниками составляет новое направление, требующее установления достоверности и истинности утверждений. Опыт построения семантических библиотек, включая LibMeta [10] и ее новую версию SciLibRu [11], показывает, что ГЗ не может быть сконструирован в отрыве от онтологии. Именно онтология предметной области определяет структуру данных, типологию понятий и семантически значимые связи, на основе которых граф строится и пополняется. Принцип «сначала онтология и тезаурус предметной области, затем граф знаний» является методологической основой такого подхода. Данные и технология их интеграции описаны в публикациях авторов [12, 13].

В настоящей работе представлена архитектура многоуровневой онтологии представления научных знаний, описания методов и их оркестрации. Оркестратор представляет собой функциональный компонент навигации по графу посредством онтологического рассуждения в многоуровневой онтологии [14].

Знания библиотек LibMeta и SciLibRu относятся к разделам математики. В работе авторов [15] предложена онтология SciLib на языке OWL/DL. Проведена материализация ГЗ на данных MathLib и Lean 4 [16]: построена таксономия доменов, выполнено сопоставление объектов с классами онтологии и сформированы мультимодальные RDF-представления для исторических и искусственных данных.

2.2. Многоуровневая онтология O: формальное описание и принципы построения

Существующие онтологические подходы к организации семантических библиотек описывают структуру научного знания: концепты предметной области, тезаурусы, термины, информационные ресурсы и их взаимосвязи [17–20]. Этого достаточно для хранения и навигации, но недостаточно для интеллектуального ассистента (программного цифрового агента), который должен выбирать и применять методы обработки данных, адаптируясь к состоянию задачи. Агент, располагающий лишь онтологией предметной области, «знает», о чем идет речь, но «не знает», как действовать и с помощью какого метода отвечать на конкретный информационный запрос [21]. Отсюда вытекает центральное требование к структуре информационной системы: *онтология системы должна описывать не только объекты мира, но и процессы работы агента, такие как методы обработки, их предусловия и постусловия, возможности агентов и правила их взаимодействия*. При таком описании оркестрация методов может быть выведена на основе правил, а не закодирована жестко в программном коде.

Предлагаемая многоуровневая онтология O строится по принципу разделения ответственности: каждый уровень описывает строго определенный аспект системы, а межуровневые отображения задают, как знания одного уровня управляют поведением на следующем [5]:

$$O = \langle L_1, \dots L_5 \rangle.$$

Каждый уровень L_k $(k = 1, \dots, 5)$ реализован как OWL 2 DL-онтология со своими классами, объектными свойствами и аксиомами. Межуровневые связи задаются отображениями $\phi_k: L_k \to L_{k+1}, (k = 1, \dots, 4)$, реализованными через именованные объектные свойства OWL. Это позволяет при выполнении рассуждений реализовать вывод через границы уровней, начав с объекта предметной области (L_1), пройти через представление (L_2), метод (L_3), агента (L_4) и достичь решения об оркестрации (L_{51}) без жестко закодированной логики маршрутизации. Нижние уровни L_1 и L_{12} описывают устойчивые (в смысле описания предметной области) объекты: концепты предметной области и информационные

ресурсы. Уровень L_3 является ядром оркестрации, он описывает методы как онтологические объекты первого уровня. Уровни L_4 и L_5 описывают динамику для агентов с их возможностями и сам процесс координации [15].

Из сказанного следуют четыре принципа, положенные в основу проектирования предлагаемой онтологии.

***Принцип* 1**. *Первичность онтологии.* Онтология первична по отношению к ГЗ. Граф есть материализация онтологии на конкретных данных предметной области, а оркестратор – функциональный компонент навигации по графу посредством онтологического рассуждения [14]. Принцип «сначала онтология и тезаурус предметной области, затем граф знаний» является методологической основой данного подхода.

***Принцип* 2**. *Разделение ответственности.* Каждый уровень онтологии описывает строго определенный аспект системы: предметную область, информационные ресурсы, методы обработки, агентные возможности, процесс оркестрации. Такое разделение восходит к архитектурному паттерну *Separation of Concerns* и обеспечивает модульность: изменение одного уровня не требует переработки остальных.

***Принцип* 3**. *Выводимость оркестрации.* Межуровневые отображения связывают уровни таким образом, что система логических рассуждений может автоматически вывести, какой метод применим к данному объекту в данном состоянии, без жестко закодированной логики маршрутизации. Это существенно отличает предлагаемый подход от традиционных API-каталогов и статических конвейеров.

***Принцип* 4**. *Единство представления.* Все уровни реализуются в рамках одного формализма OWL 2 DL. Это позволяет использовать стандартные OWL-рассуждатели (OWL Reasoner) для вывода через границы уровней и обеспечивает совместимость с существующими инструментами семантического веба (SPARQL, SHACL, SWRL).

На рис. 1 представлена иерархическая схема многоуровневой онтологии, где на каждом уровне обозначены содержание и связи в виде отображений $\phi_k: L_k \to L_{k+1} (k = 1, \dots, 4)$.

Рис. 1. Схема многоуровневой онтологии.

2.3. Уровни онтологии

2.3.1. *Первый уровень L_1: онтология предметной области.*

Первый уровень описывает структуру знания в конкретной научной области в рамках библиотеки SciLibRu, формализуя концепты, их иерархии, тематические классификаторы, формулы и тезаурусные отношения:

$$L_1 = \langle C_D, R_D, A_D, TH \rangle.$$

Здесь C_D – множество классов предметной области (*Concept, Formula, Domain, MathStatement*); R_D – объектные и аннотационные свойства, включая таксономические и содержательные отношения; A_D – аксиомы OWL 2 DL, задающие ограничения кардинальности и цепочки свойств.

Тезаурус $TH = \langle T, R_{TH} \rangle$ является неотъемлемой частью уровня: T – множество терминов, R_{TH} – иерархические и горизонтальные отношения между ними [22].

Связи между тезаурусными терминами и информационными объектами задаются семью семантически значимыми отношениями P_1–P_7. Отношения

$P_1(t, io)$ и $P_2(io, t)$ устанавливают двунаправленные связи между терминами тезауруса и информационными объектами. Отношение $P_3(r, s)$ связывает тип информационного ресурса с классом исходных объектов, $P_4(a, sa)$ – атрибут ресурса со свойством исходного класса. Отношения P_5, P_6, P_7 описывают соответственно принадлежность объекта классу источника данных, связь семантической метки с объектом и обратную связь объекта с меткой. OWL-рассуждатель расширяет ГЗ производными триплетами на основе аксиом, что и обусловило 310-кратный рост числа триплет при материализации Mathlib в SciLibRu [15, 16].

2.3.2. ***Второй уровень*** L_2***: онтология источников и ресурсов.***

Второй уровень описывает информационные ресурсы независимо от их предметного содержания. Назначение этого уровня – зафиксировать, в какой форме и из каких источников данных поступает информация в систему:

$$L_2 = \langle C_R, R_R, A_R \rangle.$$

Классы C_R включают *Publication, Monograph, Encyclopedia, Dataset, KnowledgeGraph, Corpus, Formula*. Свойства R_R описывают характеристики ресурсов: формат, язык, источник происхождения, лицензию, версии. Аксиомы A_R задают ограничения целостности, в частности, что каждый ресурс имеет не более одного канонического URI (Uniform Resource Identifier, унифицированный идентификатор ресурса).

Ключевым структурным элементом L_2 является многомодальная модель представления данных в онтологии SciLibRu. Уровень интерпретации описывает абстрактный смысл объекта, например понятие «теорема Пифагора» как математической «истины» независимо от ее конкретного выражения в виде текста или формулы. Уровень представления фиксирует конкретную форму: формальный код на Lean 4, текст на естественном языке, запись в LaTeX-нотации или визуализацию формулы. Уровень ресурса соответствует материальному носителю, то есть файлу, записи в базе данных или исполняемому коду. Все три подуровня связаны через общий URI. Такое разделение реализует принцип инвариантности: добавление новой модальности представления не изменяет интерпретационный уровень и не требует переработки L_1. Отображение $\phi_1: L_1 \to L_2$ реализуется свойством *hasRepresentation* [15].

2.3.3. Третий *уровень* L_3*: онтология методов обработки.*

Третий уровень является ядром оркестрации методов. Каждый метод семантического анализа и обработки данных представлен как самостоятельный онтологический объект с явно описанными условиями применимости. Это принципиально отличает L_3 от традиционных API-каталогов (Application Programming Interface): метод в L_3 – это не просто вызываемая процедура, а концепт с семантикой, над которым работает рассуждатель:

$$L_3 = \langle C_M, R_M, \mathrm{pre}(m_i), \mathrm{post}(m_i) \rangle.$$

Иерархия классов C_M строится от абстрактного *SemanticMethod* к конкретным подклассам: *QueryMethod* (*SPARQL, Cypher, GraphQL*), *EmbeddingMethod, GraphTraversalMethod, VerificationMethod, NLGenerationMethod, IndexingMethod*.

Свойства R_M включают *hasPrecondition, hasPostcondition, hasInputType, hasOutputType, hasComputationalCost, isComposableWith* и *conflicts*.

Каждый конкретный метод m_i формализуется как именованный индивид OWL. Его предусловие $\mathrm{pre}(m_i)$ – это конъюнкция утверждений об элементах L_1 и L_2, которые должны быть выполнены до вызова; его постусловие $\mathrm{post}(m_i)$ – это гарантированные изменения в состоянии системы после успешного исполнения. Отображение $\phi_2 \colon L_2 \to L_3$ реализуется свойством *isProcessedBy*.

2.3.4. Четвертый *уровень* L_4*: онтология агентных процессов.*

Уровень L_4 отвечает на вопрос: *кто* выполняет методы из L_3, *какими* возможностями располагает, *в каком* состоянии находится и *какую цель* преследует. Если L_3 описывает методы как абстрактные онтологические объекты с предусловиями и постусловиями, то L_4 связывает эти методы с конкретными исполнителями – программными агентами, каждый из которых обладает ограниченным набором инструментов и действует в рамках определенной политики (в рамках онтологии уровня). Разделение L_3 и L_4 реализует *Принцип* 2: добавление нового агента не требует изменения описания методов, а добавление нового метода – перепроектирования агентов, достаточно обновить множество инструментов агента.

Четвертый уровень L_4 описывает агентов как структурированные онтологические объекты, отвечая на вопрос: *кто* выполняет методы из L_3, *какими* возможностями обладает, в каком *состоянии* находится и какую *цель* преследует:

$$L_4 = \langle C_A, R_A, A_A \rangle,$$

где классы C_A включают *Agent, Capability, Tool, AgentState, Intent, Plan*.

Агент α формально задается кортежем

$$\alpha = \langle \mathrm{cap}(\alpha), \mathrm{tools}(\alpha), s_0, \pi_\alpha \rangle,$$

где $\mathrm{cap}(\alpha) \subseteq C_A$ – множество его возможностей (функций), $\mathrm{tools}(\alpha) \subseteq C_M$ – набор методов из L_3, s_0 – начальное состояние, $\pi_\alpha : S \to C_M$ – политика (правила) выбора метода.

Особую роль играют классы *Intent* и *Plan*:

– *Intent* описывает цель агента в терминах предметной области L_1 и связан с методами через свойство *requiresMethod*;

– *Plan* представляет собой частично упорядоченное множество функций с ограничениями порядка *hasPrecedence*, задающее декларативную последовательность достижения сложной цели [15, 16, 22].

2.3.5. Пятый *уровень* L_5*: онтология оркестрации.*

Уровень L_5 является метауровнем всей системы. Если L_1–L_4 описывают, *что* есть в мире (в рамках знаний системы), *в какой форме*, *как* обрабатывается и *кем* выполняется, то L_5 описывает сам *процесс координации*: как оркестратор принимает решения, как фиксирует их обоснование, как реагирует на неудачи. Ключевая функция L_5 – это обеспечение полной трассируемости цепочки рассуждений (explainability): каждое решение оркестратора привязано к конкретному SWRL-правилу, каждому состоянию до и после применения метода, каждому объекту предметной области. Без этого свойства нейросимволический агент не имеет преимуществ перед чисто нейронной системой в части объяснимости [22]:

$$L_5 = \langle C_O, R_O, A_O \rangle,$$

где классы C_O включают *OrchestratorAgent, TaskState, MethodSelection, ExecutionTrace, FallbackStrategy*. Ключевым классом является *ExecutionTrace*: каждый его

экземпляр фиксирует выбранный метод, состояние до и после его применения, а также правило онтологического вывода, обосновавшее выбор, что обеспечивает полную трассируемость цепочки рассуждений – требование *explainability*, реализованное в SciLibRu.

Пять уровней образуют связную цепочку оркестрации.

3. ОРКЕСТРАЦИЯ МЕТОДОВ

Методы, получаемые при структурировании данных, соответствуют уровням онтологического проектирования.

– На *формально-синтаксическом* уровне фиксируются объекты, заданные в формальных языках и системах доказательства: определения, леммы, теоремы, правила вывода, тактики, типы, а также их зависимости. В случае математики это может быть структура библиотеки, подобной MathLib, в которой явно представлены и формальные формулы, и дерево их использования. В других дисциплинах аналогичную роль играют формальные спецификации моделей, алгоритмов или протоколов.

– *Теоретико-структурный* уровень поднимается над конкретными формулами и описывает теории, разделы, математические и научные структуры, а также отношения между ними: обобщение и специализацию, эквивалентность формулировок, сведение одной теории к другой, импорты понятий и результатов. Здесь отдельные утверждения группируются в более крупные смысловые блоки такие, например, как теория меры, теория вероятностей, функциональный анализ, а в экспериментальных науках это определенные экспериментальные парадигмы и классы моделей.

– *Методологический* уровень фокусируется на процессуальной стороне научного знания. Он описывает методы доказательства, типичные шаблоны рассуждений, экспериментальные дизайны, протоколы сбора и анализа данных, стратегии выбора моделей и критериев. Важной частью этого уровня является описание «ролей» отдельных лемм, фактов и шагов в доказательствах или исследованиях: какие из них являются ключевыми, какие выполняют техническую вспомогательную функцию, какие обеспечивают переход между теоретическими и эмпирическими слоями.

– *Объяснительный, или дидактический*, уровень связывает всю эту формальную и структурную сложность с человеческим пониманием. На нем онтология фиксирует связи с энциклопедическими статьями, учебниками, обзорными материалами и примерами, а также задает различные стили и глубины объяснения одного и того же результата для разных аудиторий от начального уровня до экспертов. Это позволяет использовать один и тот же семантический каркас как в исследовательских, так и в образовательных сценариях.

3.1. LibMeta: оркестрация методов семантической библиотеки

Онтология LibMeta организует знания вокруг трех групп концептов: концептов, описывающих содержание предметной области и образующих тезаурус; концептов, описывающих тематические коллекции, и концептов, поддерживающих интеграцию данных из внешних источников. Тезаурус построен в соответствии со стандартом ISO 25964, включает иерархические отношения BT/NT и горизонтальное отношение RT. Граф KG MathSemanticLib строится итерационно, начиная с нулевой версии на основе Математической энциклопедии И. М. Виноградова [23]. Формулы в LibMeta являются полноправными семантическими объектами с собственными ребрами ГЗ к концептам и публикациям [9, 10, 22].

Пример навигации по предметной области обыкновенных дифференциальных уравнений иллюстрирует пайплайн с применением различных методов на рис. 2.

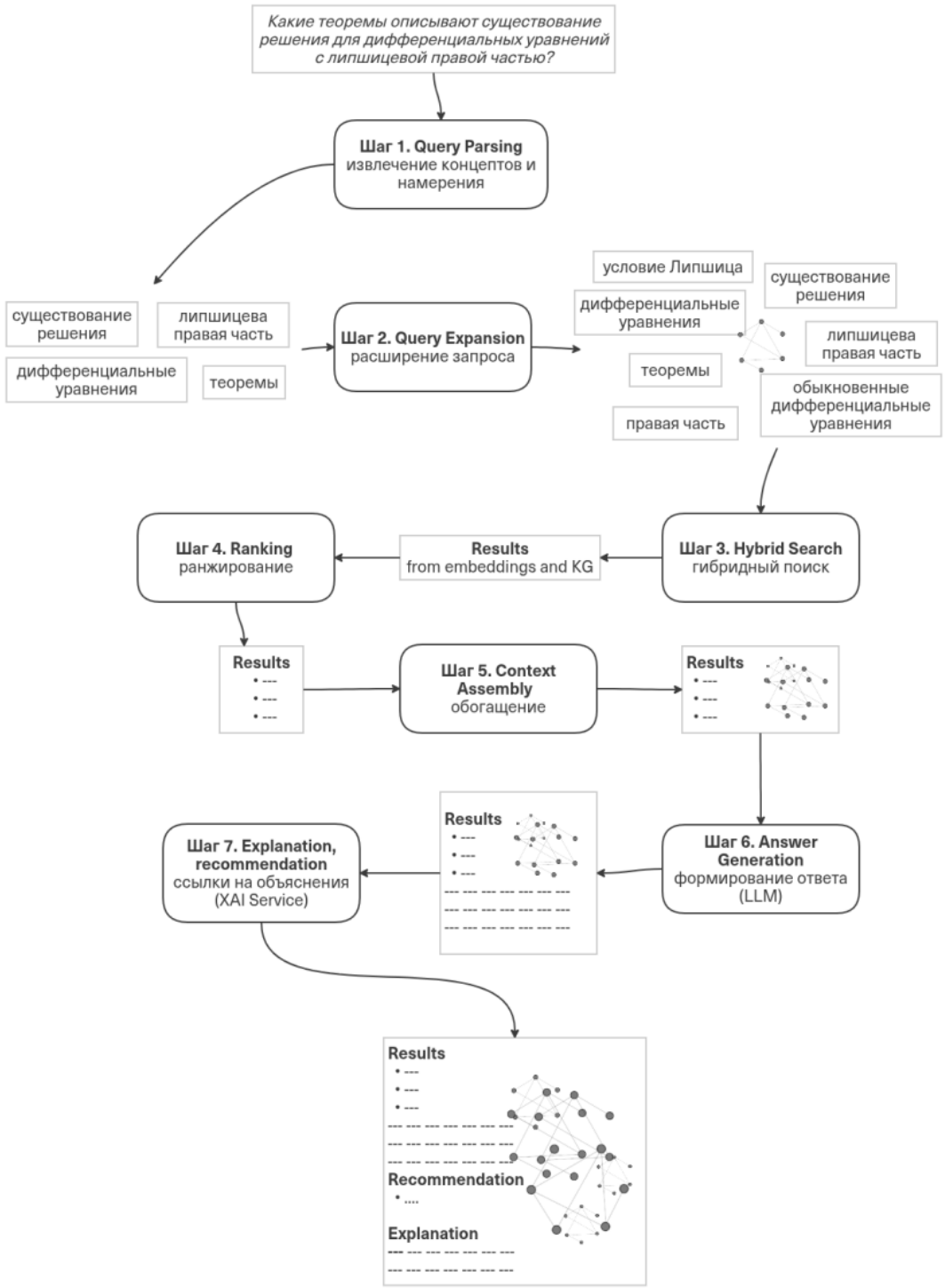


Рис. 2. Схема оркестрации методов на примере поиска в LibMeta.

При запросе «Уравнение Бернулли» оркестратор идентифицирует концепт *BernoulliODE*. Через тезаурусные связи извлекаются родственные концепты: уравнения Риккати и Якоби, коды классификации MSC, монографии и учебники

(см. рис. 2). Без онтологически управляемого контекста ведущие языковые модели (ChatGPT 4, YandexGPT) возвращают лишь общую информацию без формул и ссылок; ответ LibMeta трассируется (прослеживается в системе пошагово) до конкретных ребер графа и первоисточников [22].

Оркестрация реализуется как многошаговый пайплайн обработки запроса. На шаге разбора запроса (*query parsing*) определяются тип запроса и связанный концепт. На шаге расширения (*query expansion*) оркестратор применяет онтологические операции: подъем по ребру BT к более общему концепту или сужение по ребру NT, используя структуру тезауруса как механизм управления областью поиска, а не как классификатор пользователя. На шаге гибридного поиска (*hybrid search*) расширенный запрос обрабатывается параллельно: SPARQL-запросом к RDF-графу и векторным поиском по эмбеддингам. На шаге сборки контекста (*context assembly*) объединенный результат передается языковой модели. На шаге интеграции с XAI каждое утверждение связывается с конкретной публикацией или статьей энциклопедии через ребра графа. Это требование принципиально для математических предметных областей: недостаточно получить правильный ответ, необходимо верифицировать его через первоисточник.

3.2. SciLibRu: развитие подхода для формальной математики

Конкретным применением и развитием оркестрации методов является материализация библиотеки Lean 4 Mathlib в RDF-граф путем интеграции данных, утверждений и источников. Пайплайн материализации включает компиляцию Lean 4, извлечение метаданных парсером JIXIA, автоматическую аннотацию 660 тематических подклассов класса Domain, отображение в модель SciLib и материализацию в GraphDB. Результирующий граф содержит 66 млн RDF-троек с 6.3 млн уникальных субъектов, позволяет реализовать 310-кратный прирост относительно исходных объявлений Mathlib, достигнутый через онтологическую типизацию и материализацию выводов рассуждателя.

В SciLibRu оркестрируемые методы – это восемь режимов доставки контекста, использующих одну базовую модель DeepSeek-Prover-V2-7B и различающихся только способом формирования контекста. Первый базовый режим передает модели исходную формулировку без подсказок из графа. Второй режим

дополняет контекст леммами из векторного поиска. Следующие режимы используют граф зависимостей Mathlib с различными стратегиями обхода. Последние два гибридных режима объединяют структурный обход графа с векторным поиском и достигают наилучших результатов. На 109 трудных задачах miniF2F-Test лучший гибридный режим почти утраивает базовый показатель и вероятность того, что модель сгенерирует правильный код, с первой попытки растет с 3.56% до 10.42% [15, 24].

Принципиально важный результат заключается в том, что детерминированные символьные правила превосходят нейронные методы выбора точек входа в граф. Девять регулярных выражений, сопоставляющих структурные признаки Lean 4 с фиксированными точками входа в граф Mathlib, выполняются за 12 с и не требуют дополнительных вызовов БЯМ. Нейронный аналог требует около 30 вызовов и 134 с при более низкой точности. Этот результат имеет прямое значение для архитектуры оркестратора: для навигации по структурированному ГЗ декларативные символьные правила работают надежнее, быстрее и дешевле, чем нейронная генерация.

3.4. Общие закономерности LibMeta и SciLibRu

Анализ LibMeta и SciLibRu в их преемственности позволил выделить три устойчивые закономерности, согласованные между собой.

Первая закономерность: онтология наиболее эффективна при дефиците параметрических знаний. В LibMeta специализированные российские математические источники недоступны открытым БЯМ. В SciLibRu на легких задачах граф-расширение не дает значимого прироста; на трудных задачах, где модели не хватает «словаря» тактик, граф почти утраивает успешность. Внешнее знание ценно именно там, где внутреннего (в рамках данных и источников библиотеки) не хватает.

Вторая закономерность: символьные правила эффективнее нейронной генерации для навигации по структурированному графу. Тезаурусные операции BT/NT в LibMeta и regex-паттерны в SciLibRu являются детерминированными символьными правилами, в обоих случаях они работают быстрее и точнее. Когда пространство поиска уже структурировано онтологией, дополнительное нейронное рассуждение для навигации по нему избыточно.

Третья закономерность: трассируемость является необходимым, а не опциональным свойством. В LibMeta каждое утверждение привязано к публикациям и энциклопедическим статьям; в SciLibRu – к ребрам *usesInType/usesInValue* графа Mathlib. Без этого свойства нейросимволический агент не имеет преимуществ перед чисто нейронной системой в части объяснимости.

4. ПРИМЕР: ЦИФРОВОЙ АГЕНТ ПОСТРОЕНИЯ ШАБЛОНА ДЛЯ РЕЦЕНЗЕНТА

Рецензирование научных работ составляет одну из наиболее востребованных сфер деятельности эксперта. Это часть подготовки публикаций и докладов, которая требует времени, но в ограниченном временном промежутке, в соответствии с многочисленными дедлайнами. В классической работе [25], где перечислены 73 этапа работы с научным журналом, автор отмечает, что есть проблема «отклонение статьи». Это одно из противоречий издательского дела, поскольку статьи нужны, но нельзя пропустить ошибочные результаты и не учесть остальные особенности научных публикаций, такие как новизна, актуальность и др. Роль рецензента научных публикаций остается существенной. Тем не менее, как правило, есть определенные требования к составлению отзывов (рецензий), которые можно формализовать в виде шаблонов и далее предоставить эксперту предварительно подготовленный шаблон для анализа и работы.

Пример использования агентного подхода [26, 27] к решению проблемы рецензирования предлагается в нашем исследовании на основе оркестрации методов анализа данных (см. рис. 3). Связи ($ф_i$, $ф^{-1}_i$) отвечают за связи между уровнями онтологии при кординации действий агента. Например, пара связей (*orchestrates, getOrchestrated*) описывает выбор серии действий и проверку их результата, (*canExecute, getExecuted*) – проверку отдельных действий и их предварительных условий и проверку постусловий для них, (*isProcessedBy, getProcessedBy*) – применение отдельных методов, (*hasRepresentation, getRepresentation*) – представление извлеченных результатов с помощью этих методов.

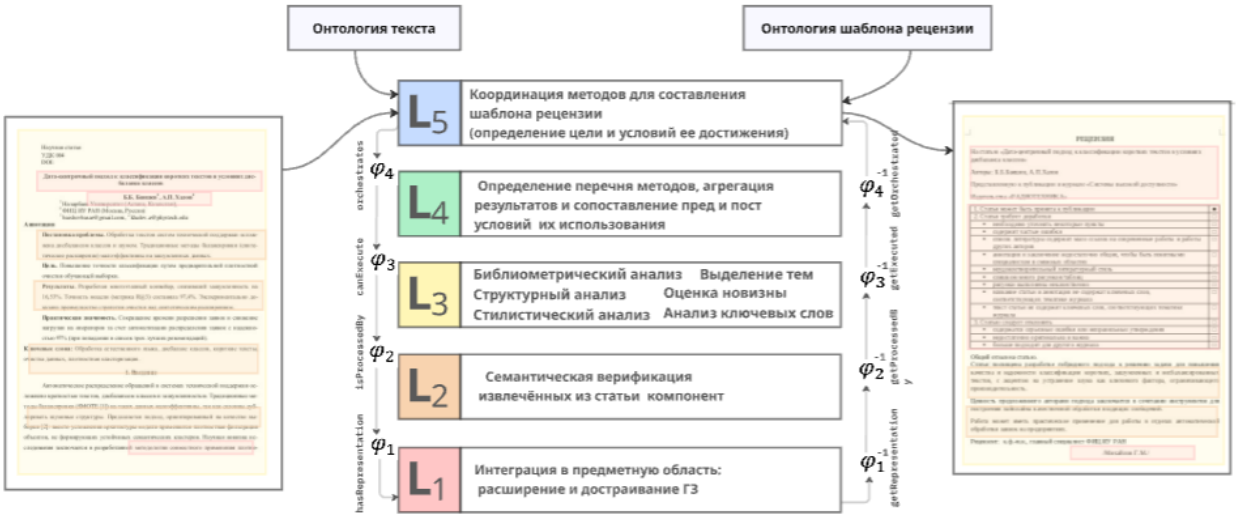


Рис. 3. Схема связей цифрового агента.

На рис. 3 показано, как шаблон рецензии ReviewTemplate заполняется поэтапно после обработки исходного текста научной статьи из раздела математических предметных областей на разных уровнях онтологии $O = \langle L_1, L_2, L_3, L_4, L_5 \rangle$:

L_1 – интеграция текста научной статьи в предметную область (используется описание предметной области, достраивается ГЗ);

L_2 – семантическая верификация (выявление метаданных, структуры текста и концептов предметной области);

L_3 – библиографическая верификация (применение методов обработки теста для структурного анализа, верификации формул, анализа ключевых слов, стилистического анализа, анализа библиографии);

L_4 – определение перечня методов (активация агента рецензирования);

L_5 – координация методов для составления шаблонов (организация конвейера для заполнения шаблона).

Результатом реализации схемы рис. 3, становится заполненная форма шаблона, которая предоставляется рецензенту для принятия решения о публикации.

ЗАКЛЮЧЕНИЕ

Идея создания многоуровневой онтологии научных данных и ее материализации в виде семантического научного графа задает основу для построения

научно-информационных систем нового поколения. В таких системах данные и знания перестают быть разрозненным набором файлов, записей и кодов; они становятся встроенными в согласованное семантическое пространство, поддерживаемое онтологией и графом знаний. Вокруг этого пространства разворачивается инфраструктура интеллектуальных сервисов, позволяющая не только находить и интегрировать информацию, но и формально рассуждать, объяснять решения ИИ-моделей и поддерживать обучение и исследовательскую работу. В результате формируется новый тип научной динамической инфраструктуры, в которой объяснимый ИИ выступает не внешней надстройкой, а частью системы, опирающейся на богатую, многоуровневую модель научного знания и ее инженерную реализацию.

Линия исследований LibMeta – SciLibRu формирует конкретную траекторию направления, которое можно обозначить как *онтологически управляемое нейросимволическое ассистирование научных исследований*. Их конвергенция к общим архитектурным принципам, трассируемости каждого решения к структуре онтологии, приоритету символьных правил для навигации по структурированному пространству, инвариантности онтологии при добавлении новых данных дает основание полагать, что эти принципы отражают фундаментальные свойства надежных нейросимволических агентов в научной среде.

СПИСОК ЛИТЕРАТУРЫ

1. PRISM. https://openai.com/ru-RU/prism/ (дата обращения 14.04.2026).

2. DoTrace. https://www.domate.ru/dotrace_platform (дата обращения 14.04.2026).

3. *Кубо Р.* Термодинамика. М.: Мир, 1970, 304 с.

4. A Unified Framework for Self-Organizing Intelligence: A Synthesis of Computational Autopoiesis, Category Theory, and Iterative Concept-Abstraction Cycles**.** Academia.edu. SSRN. 2025. https://www.academia.edu/143199301/ (дата обращения 14.04.2026).

5. *Tallam K.* From Autonomous Agents to Integrated Systems, A New Paradigm: Orchestrated Distributed Intelligence // arXiv:2503.13754. 2025. https://doi.org/10.48550/arXiv.2503.13754

6. *Bian H.* LLM-empowered knowledge graph construction: A survey // arXiv:2510.20345. 2025. https://doi.org/10.48550/arXiv.2510.20345

7. *Zabihi P., Nawara D., Ibrahim A., Kashef R.* Analyzing Bias in LLM-Augmented Knowledge Graph Systems: Taxonomy, Interaction Mechanisms, and Evaluation // Applied Sciences. 2026. Vol. 16, No. 7. Art. 3410. https://doi.org/10.3390/app16073410

8. *Schintke F. et al.* Validity constraints for data analysis workflows // Future Generation Computer Systems. 2024. Vol. 157. P. 82–97. https://doi.org/10.1016/j.future.2024.03.037

9. *Ataeva O.M., Serebraykov V.A., Tuchkova N.P.* Approaches to the organization of mathematical knowledge when forming subject thesauruses of various mathematics domains // CEUR Workshop Proc. 2018. Vol. 2260. P. 42–54. https://doi.org/10.20948/abrau-2018-66

10. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* Ontological approach to a knowledge graph construction in a semantic library // Lobachevskii J. Math. 2023. Vol. 44, No. 6. P. 2229–2239. https://doi.org/10.1134/s1995080223060471

11. *Ataeva O.M., Tuchkova N.P., Teymurazov K.B. et al.* SciLibRu, the Library of Scientific Subject Domains // Autom. Doc. Math. Linguist. 2025. Vol. 59 (Suppl. 6). P. S505–S512. https://doi.org/10.3103/S000510552570147X

12. *Кобук М.Г., Атаева О.М.* Методы семантической разметки и онтологического моделирования математических текстов в формате LaTeX // Системы высокой доступности. 2026. Т. 22, № 1. С. 90−94. https://doi.org/10.18127/j20729472-202601-18

13. *Khalov A.P., Ataeva O.M., Tuchkova N.P.* Creating a multimodal dataset for the SciLibRu semantic library using a language model // Pattern Recognit. Image Anal. 2026. 36 (In press).

14. *Стребков И.Д.* Метрические инструменты анализа графа знаний предметных областей в семантической библиотеке // Системы высокой доступности. 2026. Т. 22, № 1. С. 95−98. https://doi.org/10.18127/j20729472-202601-19

15. *Халов А.П., Атаева О.М., Тучкова Н.П.* От синтаксиса к семантике: онтология формализации научного знания SciLib // Системы высокой доступности. 2026. Т. 22, № 1. С. 65−70. https://doi.org/10.18127/j20729472-202601-13

16. *Ying H. et al.* Lean Workbook: A large-scale Lean problem set formalized from natural language math problems// arXiv:2406.03847. 2024. https://doi.org/10.48550/arXiv.2406.03847

17. *Peroni S., Shotton D.* The SPAR Ontologies // Proc. 17th Int. Semantic Web Conf. (ISWC 2018). Springer, 2018. P. 119–136.

18. *Brack A. et al.* Requirements Analysis for an Open Research Knowledge Graph // arXiv:2005.10334. arXiv:2005.10334. 2020. https://doi.org/10.48550/arXiv.2005.10334

19. *David C. et al.* Publishing Math Lecture Notes as Linked Data // Proc. CICM 2010. Springer, 2010. P. 370–375.

20. *Nevzorova O., Zhiltsov N., Kirillovich A., Lipachev E.* OntoMathPRO Ontology: A Linked Data Hub for Mathematics // Communications in Computer and Information Science. Springer, Cham, 2014. Vol. 468. P. 105–119. https://doi.org/10.1007/978-3-319-11716-4_9

21. *Masterman T., Besen, S., Sawtell M., Chao A.* The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey // arXiv:2404.11584. 2024. https://doi.org/10.48550/arXiv.2404.11584

22. *Атаева О.М., Тучкова Н.П.* Методы семантического анализа в процессах обработки данных // Системы высокой доступности. 2026. Т. 22, № 1. С. 99−104. https://doi.org/10.18127/j20729472-202601-20

23. Математическая энциклопедия. В пяти томах. Гл. ред. И. М. Виноградов. М. Советская энциклопедия (1977–1985).

24. *Shoham Y., Leyton-Brown K.* Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, 2008. 532 p.

25. *Андерсон К.* 73 этапа работы над научным журналом // Научная периодика: проблемы и решения. Т. 5 (23), сентябрь–октябрь 2014. С. 4–10. https://cyberleninka.ru/article/n/73-etapa-raboty-nad-nauchnym-zhurnalom (дата обращения 14.04.2026).

26. *Naddaf M.* AI is transforming peer review – and many scientists are worried// Nature. 2025. Vol. 639. P. 853–854. https://doi.org/10.1038/d41586-025-00894-7

27. *Farber S.* Comparing human and AI expertise in the academic peer re-

view process: towards a hybrid approach // Higher Education Research and Development. 2025. Vol. 44 (4). P. 871–885. https://doi.org/10.1080/07294360.2024.2445575

ORCHESTRATION OF METHODS OF SCIENTIFIC DATA ANALYSIS IN THE REVIEW PROCESSES

O. M. Ataeva[1] [0000-0003-0367-5575], N. P. Tuchkova[2] [0000-0001-5357-9640]

[1, 2]FRC «Computer Science and Control», Russian Academy of Sciences, Moscow, Russia

[1]oataeva@frccsc.ru, [2]ntuchkova@frccsc.ru

Abstract

This paper explores the problem of combining methods in the semantic analysis of scientific data and publications during review. At different stages of data processing in the SciLibRu system, various methods are used, a multi-level ontology is constructed, and a knowledge graph is populated, resulting in the formation of a new data structure distinct from the original. Each method individually serves its purpose in such a system, while their combined use leads to the emergence of new properties, which became the subject of this research. An example of an automatic peer review agent with explainable results is provided.

Keywords: *method orchestration, semantic analysis, domain ontology, knowledge graph, large language models, systems, categories, dynamic structures.*

REFERENCES

1. PRISM. https://openai.com/ru-RU/prism/ (date accessed: 14.04.2026).

2. DoTrace. https://www.domate.ru/dotrace_platform (date accessed: 14.04.2026).

3. *Kubo R.* Thermodynamics: An advanced course with problems and solutions. Amsterdam: North-Holland Publ. Co.; N.Y.: John Wiley and Sons, Inc., 1968. 300 p.

4. A Unified Framework for Self-Organizing Intelligence: A Synthesis of

Computational Autopoiesis, Category Theory, and Iterative Concept-Abstraction Cycles**.** Academia.edu. SSRN. 2025. https://www.academia.edu/143199301/ (date accessed: 14.04.2026).

5. *Tallam K.* From Autonomous Agents to Integrated Systems, A New Paradigm: Orchestrated Distributed Intelligence // arXiv:2503.13754. 2025. https://doi.org/10.48550/arXiv.2503.13754

6. *Bian H.* LLM-empowered knowledge graph construction: A survey // arXiv:2510.20345. 2025. https://doi.org/10.48550/arXiv.2510.20345

7. *Zabihi P., Nawara D., Ibrahim A., Kashef R.* Analyzing Bias in LLM-Augmented Knowledge Graph Systems: Taxonomy, Interaction Mechanisms, and Evaluation // Applied Sciences. 2026. Vol. 16, No. 7. Art. 3410. https://doi.org/10.3390/app16073410

8. *Schintke F. et al.* Validity constraints for data analysis workflows // Future Generation Computer Systems, 2024. Vol. 157. P. 82–97. https://doi.org/10.1016/j.future.2024.03.037

9. *Ataeva O.M., Serebraykov V.A., Tuchkova N.P.* Approaches to the organization of mathematical knowledge when forming subject thesauruses of various mathematics domains // CEUR Workshop Proc. 2018. Vol. 2260. P. 42–54. https://doi.org/10.20948/abrau-2018-66

10. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* Ontological approach to a knowledge graph construction in a semantic library // Lobachevskii J. Math. 2023. Vol. 44, No. 6. P. 2229–2239. https://doi.org/10.1134/s1995080223060471

11. *Ataeva O.M., Tuchkova N.P., Teymurazov K.B. et al.* SciLibRu, the Library of Scientific Subject Domains // Autom. Doc. Math. Linguist. 2025. Vol. 59 (Suppl 6). P. S505–S512. https://doi.org/10.3103/S000510552570147X

12. *Kobuk M.G., Ataeva O.M.* Formation of structured representations of scientific journals for integration into a knowledge graph and semantic search // Highly Available Systems. 2026. Vol. 22 (1). P. 90−94 (in Russian). https://doi.org/10.18127/j20729472-202601-18

13. *Khalov A.P., Ataeva O.M., Tuchkova N.P.* Creating a multimodal dataset for the SciLibRu semantic library using a language model // Pattern Recognit. Image Anal. 2026. 36. (In press).

14. *Strebkov I.D.* Metric tools for analyzing the knowledge graph of subject

areas in a semantic library // Highly Available Systems. 2026. 2026. Vol. 22 (1). P. 95−98 (in Russian). https://doi.org/10.18127/j20729472-202601-19

15. *Khalov A.P., Ataeva O.M., Tuchkova N.P.* От синтаксиса к семантике: онтология формализации научного знания SciLib // Highly Available Systems. 2026. Vol. 22 (1). P. 65−70. https://doi.org/10.18127/j20729472-202601-13

16. *Ying H. et al.* Lean Workbook: A large-scale Lean problem set formalized from natural language math problems // arXiv:2406.03847. 2024. https://doi.org/10.48550/arXiv.2406.03847

17. *Peroni S., Shotton D.* The SPAR Ontologies // Proc. 17th Int. Semantic Web Conf. (ISWC 2018). Springer, 2018. P. 119–136.

18. *Brack A. et al.* Requirements Analysis for an Open Research Knowledge Graph // arXiv:2005.10334. 2020. https://doi.org/10.48550/arXiv.2005.10334

19. *David C. et al.* Publishing Math Lecture Notes as Linked Data // Proc. CICM 2010. Springer, 2010. P. 370–375.

20. *Nevzorova O., Zhiltsov N., Kirillovich A., Lipachev E.* OntoMathPRO Ontology: A Linked Data Hub for Mathematics // Communications in Computer and Information Science. Springer, Cham, 2014. Vol. 468. P. 105–119. https://doi.org/10.1007/978-3-319-11716-4_9

21. *Masterman T., Besen, S., Sawtell M., Chao A.* The Landscape of Emerging AI Agent Architectures for Reasoning, Planning, and Tool Calling: A Survey // arXiv:2404.11584. 2024. https://doi.org/10.48550/arXiv.2404.11584

22. *Ataeva O.M., Tuchkova N.P.* Orchestration of semantic analysis methods // Highly Available Systems. 2026. Vol. 22 (1). P. 99−104. https://doi.org/10.18127/j20729472-202601-20 (in Russian)

23. Matematicheskaya enciklopediya. V 5 tomah. Gl. red. I. M. Vinogradov M. Sovetskaya enciklopediya (1977–1985) (in Russian).

24. *Shoham Y., Leyton-Brown K.* Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press; 2008. 532 p.

25. *Anderson K.* 73 etapa raboty nad nauchnym zhurnalom // Nauchnaya periodika: problemy i resheniya. 2014. T. 5 (23). S. 4–10 (in Russian). https://cyberleninka.ru/article/n/73-etapa-raboty-nad-nauchnym-zhurnalom (date accessed: 14.04.2026)

26. *Naddaf M.* AI is transforming peer review – and many scientists are worried// Nature. 2025. Vol. 639. P. 853–854. https://doi.org/10.1038/d41586-025-00894-7

27. *Farber S.* Comparing human and AI expertise in the academic peer review process: towards a hybrid approach // Higher Education Research and Development. 2025. Vol. 44 (4). P. 871–885. https://doi.org/10.1080/07294360.2024.2445575

СВЕДЕНИЯ ОБ АВТОРАХ

АТАЕВА Ольга Муратовна – старший научный сотрудник Вычислительного центра им. А.А. Дородницына ФИЦ ИУ РАН, к. т. н. Область научных интересов: системное программирование, базы данных, инженерия знаний и онтологии.

Olga Muratovna ATAEVA – senior researcher at the Dorodnicyn Computing Center, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences; Candidate of Technical Sciences (PhD). Research interests: systems programming, databases, knowledge engineering, ontologies. Number of scientific publications – 80.

email: oataeva@frccsc.ru

ORCID: 0000-0003-0367-5575

ТУЧКОВА Наталия Павловна – старший научный сотрудник ФИЦ ИУ РАН, кандидат физ.-мат. наук. Специалист в области алгоритмических языков и информационных технологий.

Natalia Pavlovna TUCHKOVA – senior researcher at the Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences (FRC CSC RAS) PhD in Physics and Mathematics. She is an expert in the field of algorithmic languages and information technologies.

email: NTuchkova@frccsc.ru

ORCID: 0000-0001-5357-9640

Материал поступил в редакцию 14 апреля 2026 года

УДК 004

ПОВЫШЕНИЕ УСТОЙЧИВОСТИ КЛАССИФИКАЦИИ КОРОТКИХ ТЕКСТОВ К СТОХАСТИЧЕСКОМУ ШУМУ НА ОСНОВЕ ПЛОТНОСТНОЙ ОЧИСТКИ ОБУЧАЮЩИХ ВЫБОРОК

Б. Б. Баишев[1] [0009-0007-9287-4248], А. П. Халов[2] [0009-0005-4584-8245]

[1]*Назарбаев Университет, г. Астана, Казахстан*

[2]*Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия*

[1]baishevbasar@gmail.com, [2]khalov.a@phystech.edu

Аннотация

Рассмотрена задача классификации коротких текстовых заявок в условиях значительного дисбаланса классов и зашумленности реальных потоков обращений. Показана ограниченная эффективность методов синтетического расширения выборки при работе с зашумленной разметкой. Предложен гибридный метод, сочетающий предварительную плотностную очистку данных и многоуровневое ансамблирование моделей. Применение алгоритма плотностной кластеризации позволило исключить 16.5% информационного шума от общего объема выборки. Финальная модель представлена двухуровневой архитектурой и оптимизирована с помощью байесовского поиска гиперпараметров. На отложенной тестовой выборке достигнуто значение метрики R@3, равное 97.4%. Предложенный метод позволяет автоматизировать процесс распределения заявок, существенно снижая нагрузку на операторов и сокращая время диспетчеризации обращений.

Ключевые слова: обработка естественного языка, зашумленные текстовые данные, ансамблевое обучение, робастная классификация, фильтрация шума.

ВВЕДЕНИЕ

Классификация коротких текстов является одной из фундаментальных задач современной обработки естественного языка, находящей широкое применение в автоматизации систем управления ИТ-услугами. В качестве ключевого

инструмента такой автоматизации используются методы машинного обучения, обеспечивающие интеллектуальную диспетчеризацию обращений и минимизирующие долю рутинных операций при первичной обработке данных. Несмотря на значительные успехи использования моделей глубокого обучения, задача обработки текстов из реальных пользовательских сценариев остается крайне актуальной. Для таких данных характерна высокая степень зашумленности: наличие опечаток, использование неформальной или узкоспециализированной технической лексики, а также нестандартные синтаксические структуры предложений. Эти факторы в совокупности приводят к существенной деградации метрик классификации стандартных моделей.

Особую сложность представляет сценарий, при котором обучающая выборка характеризуется распределением с «длинным хвостом» [1], что создает выраженный дисбаланс классов в сочетании с высоким уровнем шума. Эта проблема отчетливо проявляется в системах управления ИТ-услугами, где входящие обращения содержат зашумленный текст (например, *«fw:re: нужен обмен м/у базами urk/cnt/alm»*) и специфическую техническую номенклатуру (например, *«Cisco Catalyst»*, *«1C:Enterprise»*). Ввиду постоянного обновления подобных сущностей модель сталкивается с множеством внесловарных токенов, отсутствовавших на этапе обучения, что ведет к ошибкам семантического анализа. Непропорциональность классов также является критическим фактором: количество массовых типовых заявок (например, *«Сброс пароля»*) может на несколько порядков превышать число критических инцидентов в «хвосте» распределения (например, *«нет счф на номер оргтехники»*).

Известные методы борьбы с дисбалансом, такие как алгоритмы синтетического расширения выборки, например SMOTE [2], эффективно работают на искусственно созданных данных, однако демонстрируют ограниченную производительность в условиях зашумленной разметки. Во-первых, алгоритмы генерации новых примеров чувствительны к качеству исходных объектов, что ведет к тиражированию выбросов и формированию на их основе ложных искусственных кластеров. Во-вторых, искусственное выравнивание баланса искажает априорное распределение классов, приводя к общему снижению точности и полноты классификации.

В настоящей работе предложен альтернативный подход, основанный на концепции приоритета качества обучающих данных [3]. Смещая фокус с усложнения архитектур нейросетевых моделей на повышение репрезентативности исходной выборки, мы используем стратегию плотностной фильтрации. Этот метод позволяет эффективно исключать шумовые объекты, находящиеся в разреженных областях признакового пространства и не формирующие устойчивых семантических кластеров. Научная новизна исследования заключается в разработке комплексной методологии совместного применения плотностной очистки данных и ансамблевых методов, что обеспечивает экспериментально подтвержденный прирост точности классификации по сравнению с базовыми подходами.

ФОРМАЛЬНАЯ ПОСТАНОВКА ЗАДАЧИ

Задача формулируется как многоклассовая классификация текстовых сообщений в условиях зашумленности разметки и признакового пространства.

Пусть $D = \{(x_i, y_i)\}_{i=1}^{N}$ — обучающая выборка, где объект $x_i = (t_i, m_i)$ включает неструктурированное текстовое описание t_i и вектор метаданных m_i .

Целевая переменная $y_i \in C = \{c_1, \ldots, c_k\}$ соответствует одной из K групп поддержки. Специфика предметной области (домена) накладывает на множество D три ключевых ограничения.

Сверхкраткость векторов: средняя длина $|t_i| \leq 7$ токенов, что приводит к высокой разреженности признакового пространства и дефициту контекстной информации.

Классовый дисбаланс: распределение классов $P(y)$ имеет «тяжелый хвост». Коэффициент дисбаланса, определяемый как $\rho = \frac{\max(N_c)}{\min(N_c)}$, достигает значений $\rho > 1000$. Значительная часть классов представлена малым количеством примеров ($n_c < 10$), недостаточным для обучения параметрических моделей без предварительной аугментации или регуляризации.

Стохастический шум: существует подмножество $D_{\text{noise}} \subset D$, для которого истинная метка y_i присвоена ошибочно вследствие человеческого фактора, либо текст t_i не содержит семантической информации, релевантной для задачи классификации.

Целью настоящей работы является построение отображения $f: X \to C$, которое минимизирует функцию потерь на тестовой выборке. Основной метрикой качества выбрана $\mathrm{R@k}$, так как в прикладном сценарии критически важно наличие истинного класса в списке из K наиболее вероятных рекомендаций системы.

ОБЗОР АНАЛОГИЧНЫХ ИССЛЕДОВАННИЙ

Современные подходы к классификации коротких текстов эволюционировали от базовых частотных методов, таких как TF-IDF [4], к использованию плотных векторных представлений в сочетании с ансамблевыми алгоритмами. Исследования подтверждают высокую эффективность комбинации нейросетевых кодировщиков и алгоритмов градиентного усиления [5], а также многоуровневой композиции классификаторов на базе решающих деревьев [6]. Однако в домене систем управления ИТ-услугами итоговая точность глубоких моделей критически зависит от качества предобработки и устойчивости к «шумным» классам [7].

Проблема дисбаланса, типичная для журналов регистрации событий [8], существенно ограничивает применимость стандартных подходов. В работе [9] показано, что генерация искусственных примеров на коротких зашумленных текстах часто искажает семантику и не превосходит тривиальное дублирование [9].

В качестве альтернативы активно исследуется алгоритмическая фильтрация данных, в частности выявление структурных аномалий с помощью плотностной кластеризации HDBSCAN [10, 11] поверх стандартных векторных представлений. Существенным ограничением таких решений является использование «замороженных» общелексических моделей, не способных формировать корректные векторные представления для специфического ИТ-сленга. В настоящей работе этот пробел устраняется за счет интеграции модели [12], адаптированной к предметной области (доменно-адаптированной), что обеспечивает семантически корректную фильтрацию шума и формирование качественного признакового пространства.

МЕТОДЫ

Для решения формализованной задачи классификации в условиях «длинного хвоста» и шума разработан многоступенчатый конвейер, основанный на концепции приоритета качества данных. Архитектура включает три последовательных этапа: предварительную подготовку и квотирование данных, плотностную фильтрацию признакового пространства и многоуровневое ансамблирование моделей.

Подготовка и балансировка данных

Для формирования обучающей выборки из исходного корпуса со значительным дисбалансом был применен алгоритм квотирования, основанный на степенном сглаживании частот. Целевой размер выборки Q_c для класса C рассчитывался следующим образом:

$$Q_c = \text{clip}\left(N_{\text{total}} \frac{(N_c)^{\alpha}}{\sum_j (N_j)^{\alpha}}, L_{\min}, Rm_{\text{ref}}\right),$$

где N_c – исходное количество примеров класса, clip – функция усечения, ограничивающая вычисляемое значение заданным диапазоном, α – коэффициент сглаживания, увеличивающий вес редких классов, $L_{\min}$ – нижний порог, гарантирующий корректность перекрестной проверки, Rm_{ref} – верхний порог, в котором R ограничивает отношение преобладающего класса к медианному значению m_{ref}. Для исключения смещения в сторону крупных клиентов квота заполнялась стратифицированно. Количество примеров $n_{c,s}$, отбираемых от конкретного источника s для класса c, определялось пропорционально его доле в исходных данных:

$$n_{c,s} \propto Q_c \ \frac{N_{c,s}}{N_c},$$

где $N_{c,s}$ – исходное количество заявок класса c от источника s. Это обеспечивает репрезентативность выборки и предотвращает переобучение модели на специфической лексике одного заказчика.

На этапе лексической очистки из текстов заявок удалялась нерелевантная информация: IP-адреса и URL-ссылки заменялись на специальные токены <ip>

и <url> соответственно. Было также произведено удаление неразрывных пробелов, невидимых символов Unicode и приведение текста к нижнему регистру. Для снижения размерности словаря и исключения утечки данных между обучающей и валидационной выборками реализован двухэтапный поиск дубликатов. Поиск **точных дубликатов** осуществлялся путем удаления записей с идентичным хеш-значением алгоритма SHA-1, вычисленным от нормализованного текста. Для выявления **нечетких дубликатов** (семантически близких заявок, отличающихся опечатками или автогенерируемыми метками времени) применялся алгоритм SimHash [15]. Вектор признаков для хеширования формировался на основе символьных n-грамм ($n \in \{3, 4, 5\}$), что обеспечивает устойчивость к незначительным изменениям в тексте. Пороговое значение расстояния Хэмминга для определения дубликата было установлено на уровне $d \leq 3$ бит.

Плотностная фильтрация

Процедура очистки данных реализуется через последовательность трех шагов.

Векторизация. Для преобразования текстов в векторное пространство использован доменно-адаптированный кодировщик на базе архитектуры XLM-RoBERTa Large [12]. Модель принимает на вход нормализованный текст t_i, а в качестве векторного представления заявки v_i используется скрытое состояние специального токена [CLS] последнего скрытого слоя:

$$v_i = h_{[\mathrm{CLS}]}\,.$$

Снижение размерности. Для повышения плотности кластеров перед подачей в алгоритм кластеризации размерность векторов была снижена с 1024 до 64 компонент методом главных компонент.

Плотностная кластеризация. Разделение на семантические группы производилось алгоритмом HDBSCAN [11]. Данный метод позволяет автоматически определять количество кластеров на основе плотности распределения и явно выделять объекты, находящиеся в разреженных областях.

По итогам работы алгоритма объекты, получившие метку шума, интерпретировались как стохастические аномалии (нерелевантные заявки, спам, редкие выбросы) и исключались из обучающей выборки.

Ансамблевая классификация

Финальный этап конвейера отвечает за построение пространства признаков и обучение двухуровневого ансамбля моделей.

Конструирование признаков. Процедура включает в себя семантическое обогащение коротких текстов путем конкатенации структурированных метаданных (тегов иерархии, меток длины и флагов детализации). Для повышения разделяющей способности сгенерированы доменные ключевые слова. С помощью критерия χ^2 выделены токены с максимальной предсказательной силой для каждого класса. Специфика источников данных учитывается через формирование профиля клиента с использованием кодирования средним значением целевой переменной [15]. Профиль включает вектор априорных вероятностей, коэффициент доминирования преобладающего класса и информационную энтропию распределения заявок $H = -\sum p_i \log p_i$. Дополнительно генерируется вектор базовых инженерных метрик: логарифмированные длины, коэффициент лексического разнообразия.

Базовые модели первого уровня. Для формирования семантического пространства решений использован гибридный подход. За фиксацию лексических паттернов отвечают две модели логистической регрессии, обученные на символьных и словных n-граммах с TF-IDF векторизацией (с применением сублинейного масштабирования). Для извлечения глубоких семантических признаков использован трансформер XLM-RoBERTa Large, предварительно адаптированный на корпусе технических текстов [12]. С учетом наличия субъективного шума в разметке инцидентов обучение нейросетевой модели производилось с применением сглаживания меток для предотвращения переобучения на ошибочных примерах:

$$y_{\text{target}} = (1 - \varepsilon) y_{\text{true}} + \frac{\varepsilon}{K},$$

где ε – коэффициент сглаживания, K – общее количество классов.

Архитектура мета-классификатора. Итоговое решение формируется алгоритмом градиентного усиления XGBoost [16] в парадигме многоуровневого ансамблирования [13]. Для предотвращения утечки данных вероятностные прогнозы базовых моделей генерируются строго методом перекрестного прогнози-

рования на отложенных блоках выборки. Поскольку алгоритмы на основе деревьев решений эффективнее работают с признаками, линейно разделяемыми на интервале $(-\infty, +\infty)$, было применено обратное сигмоидальное преобразование вероятностей p в пространство логарифмов отношения шансов:

$$z = \ln \frac{p}{1-p}.$$

Итоговый вектор признаков x_{meta} для обучения финальной мета-модели формируется путем объединения:

$$x_{\text{meta}} = [z_{\text{RoBERTa}} \oplus z_{\text{CharLR}} \oplus z_{\text{WordLR}} \oplus x_{\text{client}} \oplus x_{\text{domain}} \oplus x_{\text{extra}}],$$

где $\oplus$ обозначает операцию векторного сцепления преобразованных прогнозов базовых моделей (z), профиля клиента (x_{client}), доменных (x_{domain}) и инженерных (x_{extra}) признаков.

ЭКСПЕРИМЕНТЫ

Набор данных и базовые стратегии

Настоящее исследование проводилось на закрытом корпоративном корпусе системы технической поддержки, содержащем 570 тыс. текстовых записей. Исходный дисбаланс классов достигал значения коэффициента 220. Применение разработанного алгоритма адаптивного квотирования сократило обучающую выборку до 55 тыс. объектов, снизив дисбаланс до 14.6 при сохранении репрезентативности редких классов. Распределение классов до и после квотирования представлено на рис. 1.

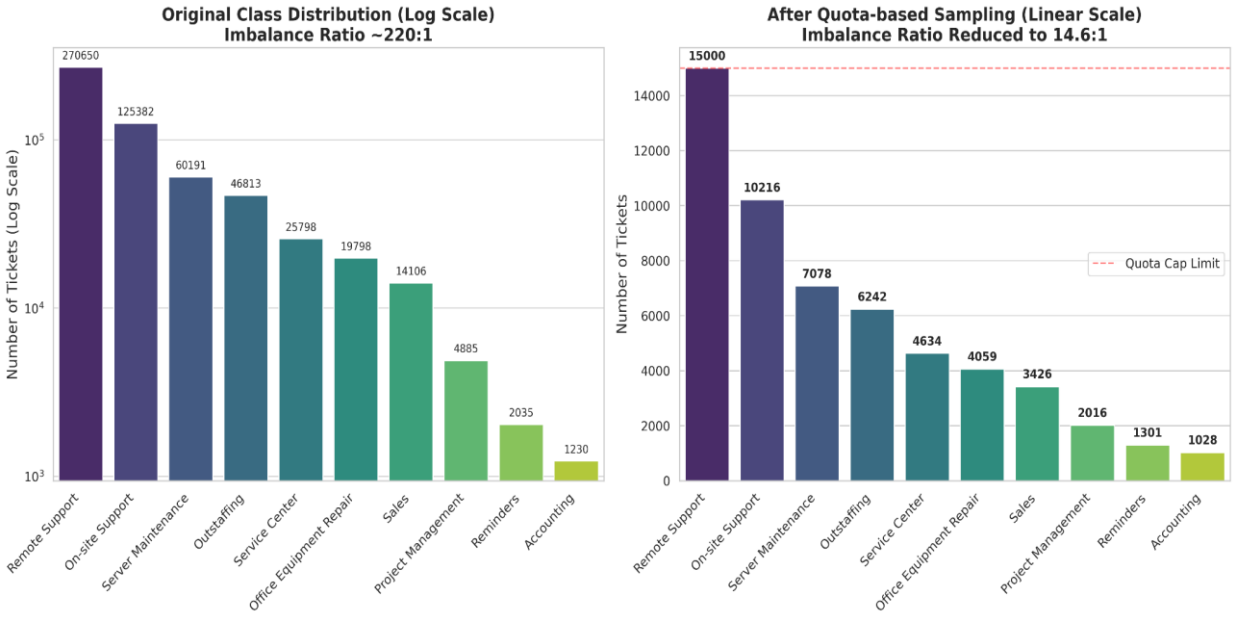


Рис. 1. Распределение классов до и после квотирования.

Далее обоснуем стратегию очистки**.** Для определения оптимального метода работы с шумом сравнивались три подхода к подготовке данных: синтетическое расширение (SMOTE), базовый подход (обучение на данных, сбалансированных методом квотирования, без удаления шума) и предложенная плотностная фильтрация (HDBSCAN). Базовым классификатором выступал алгоритм градиентного усиления. Сравнительный анализ (рис. 2) подтвердил негативное влияние синтетического расширения на зашумленных данных: метрика точности снизилась из-за тиражирования ошибок разметки. Напротив, плотностная фильтрация повысила точность на 1.99% и F1-меру на 2.57% относительно базового подхода, доказав, что удаление стохастического шума эффективнее искусственного увеличения выборки. Необходимо отметить, что плотностная кластеризация является наиболее ресурсоемким этапом конвейера, однако она проводится однократно на этапе подготовки данных и выполняется на центральном процессоре. Это исключает аппаратную зависимость от графических ускорителей и обеспечивает высокую универсальность предложенного решения при внедрении.

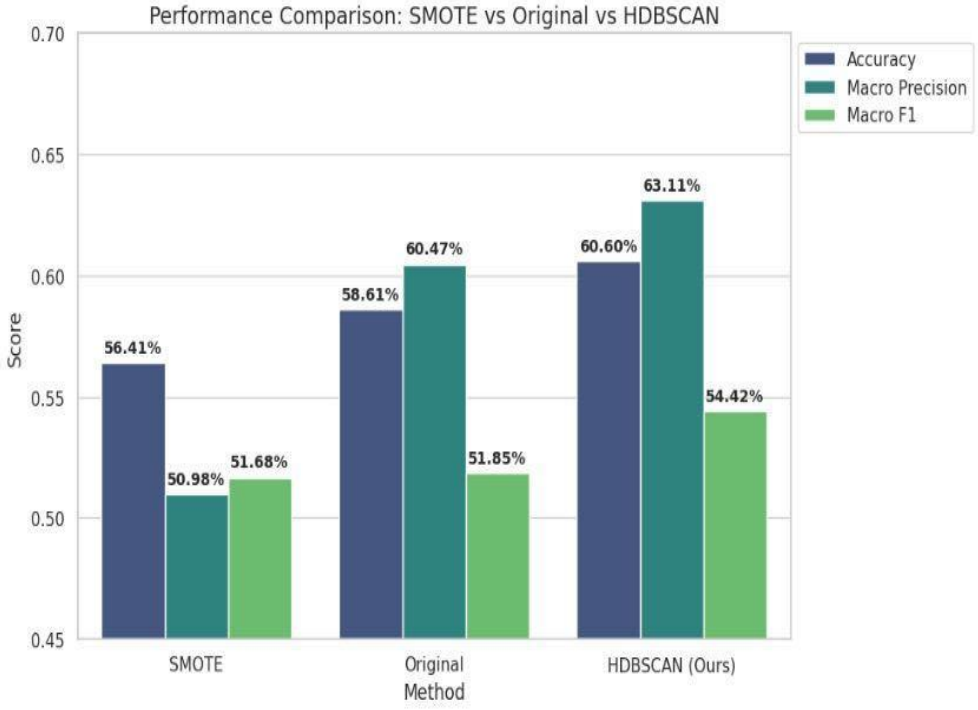


Рис. 2. Сравнение влияния различных методов предобработки на метрики классификации.

Оценка ансамбля и оптимизация

В качестве базовых моделей первого уровня обучались логистические регрессии на символьных и словных n-граммах, а также доменно-адаптированный трансформер. Оценка производилась методом пятикратной перекрестной проверки. Как видно из табл. 1, линейные модели продемонстрировали качество, сопоставимое с нейросетевым подходом, обеспечив при этом необходимую независимость предсказаний для успешного ансамблирования.

Табл. 1. Результаты базовых моделей первого уровня.

Режим	Признаки	Точность
A	Char-level TF-IDF + LogReg	0.654 ± 0.005
B	Word-level TF-IDF + LogReg	0.645 ± 0.005
C	XLM-RoBERTa Large	0.691 ± 0.006

Оптимизация мета-классификатора. Векторы вероятностей базовых моделей агрегировались с мета-признаками для обучения финальной мета-модели. Применение алгоритма байесовской оптимизации гиперпараметров [17] позволило улучшить обобщающую способность ансамбля (табл. 2).

Табл. 2. Сравнение производительности ансамбля

Конфигурация	Точность	Макро-F1	ROC-AUC
Default XGBoost	0.765 ± 0.004	0.701	0.953
Optuna Tuned	0.767 ± 0.004	0.706	0.954

График функции потерь на валидационных выборках демонстрирует высокую устойчивость алгоритма. Выход кривой на асимптотическое плато без последующего роста свидетельствует об эффективной работе механизма ранней остановки (рис. 3).

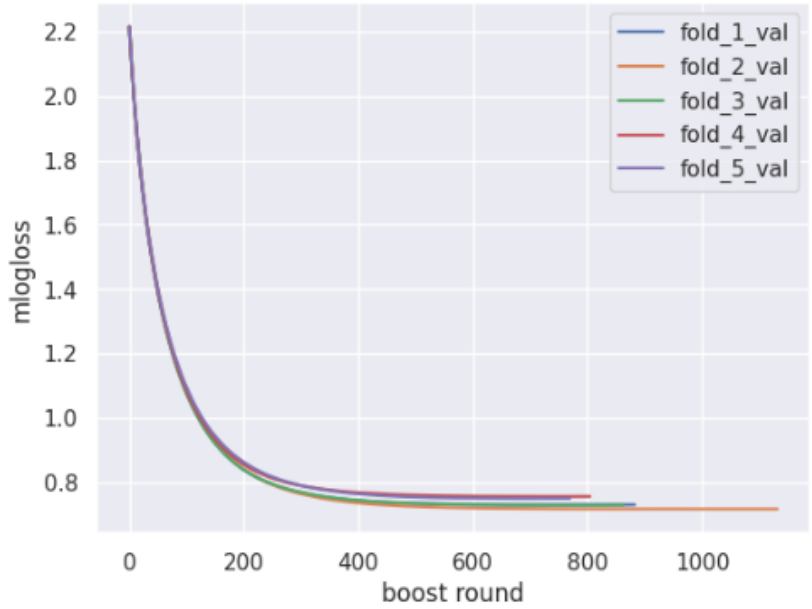


Рис. 3. График функции потерь на валидационных выборках.

Итоговая оценка качества

Для валидации применимости решения в промышленной эксплуатации проводилось тестирование на отложенной выборке из 15 тыс. заявок, имитирующей реальный поток обращений. В качестве ключевых метрик использовались точность первого выбора (R@1) и точность по трем лучшим предсказаниям (R@3), отражающая эффективность системы как помощника оператора. Сравнительный анализ сценариев эксплуатации представлен в табл. 3. В идеальных условиях (клиенты, известные системе, очищенные данные) метрика R@1 составила 81.7%, а R@3 достигла 97.4%. Важно отметить, что даже в наиболее сложном сценарии, включающем зашумленные данные и запросы от новых заказчиков, метрики сохранили высокие значения: R@1 = 72.5%, R@3 = 91.5%. Это подтверждает применимость модели в промышленной эксплуатации: более чем в 90% случаев верное решение находится среди трех предложенных рекомендаций.

Табл. 3. Сравнительный анализ производительности.

Сценарий	Покрытие	Полнота R@1	Полнота R@3	Взвешенная F1-мера	ROC-AUC
Известные клиенты, без шума	84.31%	0.817	0.974	0.821	0.961
Известные/новые клиенты, без шума	84.0%	0.773	0.942	0.794	0.951
Известные клиенты, с шумом	100%	0.764	0.943	0.762	0.923
Известные/новые клиенты, с шумом	100%	0.725	**0.915**	0.735	0.908

ЗАКЛЮЧЕНИЕ

Предложен комплексный метод автоматической классификации коротких текстов для систем технической поддержки, функционирующих в условиях силь-

ного классового дисбаланса и высокого уровня стохастического шума. Экспериментально доказано преимущество алгоритмов плотностной фильтрации над традиционными методами синтетического расширения выборки. В частности, применение кластеризации HDBSCAN позволило выявить и исключить 16.5% зашумленных объектов, что предотвратило тиражирование ошибок разметки и обеспечило значимый прирост метрик качества по сравнению с алгоритмом SMOTE. Полученные результаты подтверждают фундаментальную гипотезу: при работе с реальными корпоративными данными качество обучающей выборки приоритетнее ее объема.

Итоговая архитектура решения, реализованная в парадигме многоуровневого ансамблирования и настроенная с помощью байесовской оптимизации гиперпараметров, объединила глубокие семантические, лексические и интерпретируемые мета-признаки. Ансамблевая модель продемонстрировала высокую надежность: на отложенной тестовой выборке метрика R@3 достигла 97.4%. Данный результат позволяет эффективно использовать разработанный классификатор в контуре промышленной эксплуатации как систему поддержки принятия решений, где вероятность ошибки в рекомендациях составляет менее 3%. Разработанный вычислительный конвейер отличается высокой степенью универсальности и не имеет жесткой привязки к специфике исходной выборки. В связи с этим перспективным направлением для дальнейших исследований является валидация предложенной методологии на текстовых корпусах из смежных технических предметных областей.

СПИСОК ЛИТЕРАТУРЫ

1. *Zhang Y. et al.* Deep Long-Tailed Learning: A Survey // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023. Vol. 45, No. 3. P. 3079–3099. https://doi.org/10.1109/TPAMI.2021.3114116

2. *Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P.* SMOTE: synthetic minority over-sampling technique // Journal of Artificial Intelligence Research. 2002. Vol. 16. P. 321–357. https://doi.org/10.1613/jair.953

3. *Zha D. et al.* Data-centric Artificial Intelligence: A Survey // ACM Computing Surveys. 2025. Vol. 57, No. 5. Article 129. https://doi.org/10.1145/3711118

4. *Salton G., Buckley C.* Term-weighting approaches in automatic text retrieval // Information Processing & Management. 1988. Vol. 24, No. 5. P. 513–523. https://doi.org/10.1016/0306-4573(88)90021-0

5. *Batiuk T., Dosyn D.* Intellectual analysis of textual data in social networks using BERT and XGBOOST // Vìsnik Nacìonal'nogo Unìversitetu L'vìvs'ka Polìtehnìka Serìâ Ìnformacìjnì Sistemi Ta Merežì. 2025. Vol. 17. P. 44–60. https://doi.org/10.23939/sisn2025.17.044

6. *Parmar M., Tiwari A.* Enhancing text classification performance using stacking ensemble method with TF-IDF feature extraction // Proceedings of the 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI). Kathmandu, Nepal. 2024. P. 166–174. https://doi.org/10.1109/ICMCSI61480.2024.10493890

7. *Zemp M.* Text classification of service desk tickets. Master's thesis. Winterthur, Zurich University of Applied Sciences. 2021. https://www.zhaw.ch/storage/shared/upload/MAS21_Ticket_Classification_Zemp.pdf (дата обращения: 12.02.2026)

8. *Akhbardeh F., Alm C.O., Zampieri M., Desell T.* Handling extreme class imbalance in technical logbook datasets // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP). Online. 2021. P. 4034–4045. https://doi.org/10.18653/v1/2021.acl-long.312

9. *Padurariu C., Breaban M.E.* Dealing with data imbalance in text classification // Procedia Computer Science. 2019. Vol. 159. P. 736–745. https://doi.org/10.1016/j.procs.2019.09.229

10. *Asyaky M.S., Mandala R.* Improving the performance of HDBSCAN on short text clustering by using word embedding and UMAP // 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA). Bandung, Indonesia. 2021. P. 1–6. https://doi.org/10.1109/ICAICTA53211.2021.9640285

11. *McInnes L., Healy J., Astels S.* hdbscan: Hierarchical density based clustering // Journal of Open Source Software. 2017. Vol. 2, No. 11. P. 205. https://doi.org/10.21105/joss.00205

12. *Халов А.П., Атаева О.М.* Автоматические и полуавтоматические методы построения графа знаний предметной области и расширения онтологии // Электронные библиотеки. 2025. Т. 28, № 6. С. 1481–1519. https://doi.org/10.26907/1562-5419-2025-28-6-1481-1519

13. *Wolpert D.H.* Stacked generalization // Neural Networks. 1992. Vol. 5, No. 2. P. 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

14. *Charikar M.S.* Similarity estimation techniques from rounding algorithms // Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC). 2002. P. 380–388. https://doi.org/10.1145/509907.509965

15. *Micci-Barreca D.* A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems // SIGKDD Explorations Newsletter. 2001. Vol. 3, No. 1. P. 27–32. https://doi.org/10.1145/507533.507538

16. *Chen T., Guestrin C.* XGBoost: A scalable tree boosting system // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). San Francisco, USA. 2016. P. 785–794. https://doi.org/10.1145/2939672.2939785

17. *Akiba T., Sano S., Yanase T., Ohta T., Koyama M.* Optuna: A next-generation hyperparameter optimization framework // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). Anchorage, USA. 2019. P. 2623–2631. https://doi.org/10.1145/3292500.3330701

IMPROVING SHORT TEXT CLASSIFICATION ROBUSTNESS TO STOCHASTIC NOISE BASED ON DENSITY-DRIVEN TRAINING DATA CLEANING

B. B. Baishev[1] [0009-0007-9287-4248], A. P. Khalov[2] [0009-0005-4584-8245]

[1]*Nazarbayev University, Astana, Kazakhstan*

[2]*Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia*

[1]baishevbasar@gmail.com, [2]khalov.a@phystech.edu

Abstract

The paper addresses the problem of short text request classification under conditions of significant class imbalance and high noise levels in real-world communication flows. The limited effectiveness of synthetic oversampling techniques when dealing with noisy labeling is demonstrated. A hybrid method is proposed, combining preliminary density-based data cleaning and multi-level model ensembling. The application of a density-based clustering algorithm enabled the exclusion of 16.5% of informational noise from the total sample volume. The final model features a two-level architecture and is optimized using Bayesian hyperparameter search. A Recall@3 (R@3) metric of 97.4% was achieved on a hold-out test set. The proposed method allows for the automation of the request distribution process, significantly reducing operator workload and decreasing dispatch time.

Keywords*: natural language processing, noisy text data, ensemble learning, robust classification, noise filtering.*

REFERENCES

1. *Zhang Y. et al.* Deep Long-Tailed Learning: A Survey // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023. Vol. 45, No. 3. P. 3079–3099. https://doi.org/10.1109/TPAMI.2021.3114116

2. *Chawla N.V., Bowyer K.W., Hall L.O., Kegelmeyer W.P.* SMOTE: synthetic minority over-sampling technique // Journal of Artificial Intelligence Research. 2002. Vol. 16. P. 321–357. https://doi.org/10.1613/jair.953

3. *Zha D. et al.* Data-centric Artificial Intelligence: A Survey // ACM Computing Surveys. 2025. Vol. 57, No. 5. Article 129.
https://doi.org/10.1145/3711118

4. *Salton G., Buckley C.* Term-weighting approaches in automatic text retrieval // Information Processing & Management. 1988. Vol. 24, No. 5. P. 513–523.
https://doi.org/10.1016/0306-4573(88)90021-0

5. *Batiuk T., Dosyn D.* Intellectual analysis of textual data in social networks using BERT and XGBOOST // Vìsnik Nacìonal'nogo Unìversitetu L'vìvs'ka Polìtehnìka Serìâ Ìnformacìjnì Sistemi Ta Merežì. 2025. Vol. 17. P. 44–60.
https://doi.org/10.23939/sisn2025.17.044

6. *Parmar M., Tiwari A.* Enhancing text classification performance using stacking ensemble method with TF-IDF feature extraction // Proceedings of the 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI). Kathmandu, Nepal. 2024. P. 166–174.
https://doi.org/10.1109/ICMCSI61480.2024.10493890

7. *Zemp M.* Text classification of service desk tickets. Master's thesis. Winterthur, Zurich University of Applied Sciences. 2021.
https://www.zhaw.ch/storage/shared/upload/MAS21_Ticket_Classification_Zemp.pdf

8. *Akhbardeh F., Alm C.O., Zampieri M., Desell T.* Handling extreme class imbalance in technical logbook datasets // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP). Online. 2021. P. 4034–4045.
https://doi.org/10.18653/v1/2021.acl-long.312

9. *Padurariu C., Breaban M.E.* Dealing with data imbalance in text classification // Procedia Computer Science. 2019. Vol. 159. P. 736–745.
https://doi.org/10.1016/j.procs.2019.09.229

10. *Asyaky M.S., Mandala R.* Improving the performance of HDBSCAN on short text clustering by using word embedding and UMAP // 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA). Bandung, Indonesia. 2021. P. 1–6.
https://doi.org/10.1109/ICAICTA53211.2021.9640285

11. *McInnes L., Healy J., Astels S.* hdbscan: Hierarchical density based clustering // Journal of Open Source Software. 2017. Vol. 2, No. 11. P. 205. https://doi.org/10.21105/joss.00205

12. *Khalov A.P., Ataeva O.M.* Automatic and semi-automatic methods for constructing a domain knowledge graph and ontology expansion // Russian Digital Libraries Journal. 2025. Vol. 28, No. 6. P. 1481–1519 (in Russian). https://doi.org/10.26907/1562-5419-2025-28-6-1481-1519

13. *Wolpert D.H.* Stacked generalization // Neural Networks. 1992. Vol. 5, No. 2. P. 241–259. https://doi.org/10.1016/S0893-6080(05)80023-1

14. *Charikar M.S.* Similarity estimation techniques from rounding algorithms // Proceedings of the thirtieth annual ACM symposium on Theory of computing (STOC). 2002. P. 380–388. https://doi.org/10.1145/509907.509965

15. *Micci-Barreca D.* A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems // SIGKDD Explorations Newsletter. 2001. Vol. 3, No. 1. P. 27–32. https://doi.org/10.1145/507533.507538

16. *Chen T., Guestrin C.* XGBoost: A scalable tree boosting system // Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). San Francisco, USA. 2016. P. 785–794. https://doi.org/10.1145/2939672.2939785

17. *Akiba T., Sano S., Yanase T., Ohta T., Koyama M.* Optuna: A next-generation hyperparameter optimization framework // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). Anchorage, USA. 2019. P. 2623–2631. https://doi.org/10.1145/3292500.3330701

СВЕДЕНИЯ ОБ АВТОРАХ

БАИШЕВ Басар Бауржанович – студент 2-го курса, ассистент-исследователь, Назарбаев Университет, кафедра «Компьютерные науки». Область научных интересов: обработка естественного языка (NLP), машинное обучение, нейронные сети, анализ несбалансированных данных.

Bassar Baurzhanovich BAISHEV – second-year undergraduate student, research assistant at the Department of Computer Science Nazarbayev University. Research interests: natural language processing (NLP), machine learning, neural networks, imbalanced data analysis.

email: baishevbasar@gmail.com

ORCID: 0009-0007-9287-4248

***ХАЛОВ* Андрей Петрович** – аспирант МФТИ (ФПМИ), кафедра «Интеллектуальные системы». Область научных интересов: онтологическое моделирование, графы знаний, извлечение знаний из текстов (NER/RE, RAG), многоагентные системы и планирование, применение LLM в корпоративных ИС.

Andrey Petrovich KHALOV – PhD student at the Moscow Institute of Physics and Technology (MIPT), Phystech School of Applied Mathematics and Informatics, Department of Intelligent Systems. Research interests: ontological modeling, knowledge graphs, information extraction from text (NER/RE, RAG), multi-agent systems and planning, application of LLMs in enterprise information systems.

email: khalov.a@phystech.edu

ORCID: 0009-0005-4584-8245

Материал поступил в редакцию 24 марта 2026 года

УДК 004.912

МЕТОДЫ АВТОМАТИЧЕСКОГО ПРИСВОЕНИЯ КОДОВ УДК МАТЕМАТИЧЕСКИМ СТАТЬЯМ: ОЦЕНКА КЛАССИЧЕСКИХ И НЕЙРОСЕТЕВЫХ ПОДХОДОВ

Б. Т. Гизатуллин[1] [0009-0000-6251-9260], О. А. Невзорова[2] [0000-0001-8116-9446]

[1, 2]Казанский (Приволжский) федеральный университет, г. Казань, Россия

[1]gizat.blt@gmail.com, [2]onevzoro@gmail.com

Аннотация

Универсальная десятичная классификация (УДК) – это иерархическая система индексирования, в рамках которой одной публикации могут соответствовать один или несколько кодов. Ручное присвоение кодов УДК трудоемко и нередко оказывается неоднородным. В работе рассмотрена задача автоматического присвоения кодов УДК русскоязычным математическим статьям. Цель исследования – сравнить различные сочетания текстовых представлений и моделей классификации на едином корпусе и определить наиболее эффективные конфигурации. Для этого был сформирован корпус из 4194 статей с ресурса Math-Net.Ru, включающий полные тексты, аннотации, метаданные и коды УДК; были выполнены извлечение текста из PDF-файлов, очистка артефактов верстки и нормализация кодов. В эксперименте сопоставлялись текстовые представления TF-IDF, Word2Vec, SciRus-tiny и SciRus-tiny3.5 в сочетании с моделями логистической регрессии, Complement Naive Bayes (CNB) и CatBoost. Наилучшие результаты в обеих постановках – однозначной (single-label) и многозначной (multi-label) – показала модель TF-IDF + LogReg; близкие результаты продемонстрировала конфигурация TF-IDF + CNB. Полученные результаты могут быть использованы при разработке систем автоматической рубрикации научных публикаций, рекомендательных сервисов для авторов и редакторов, а также средств контроля качества тематической разметки.

Ключевые слова: *автоматическая классификация, универсальная десятичная классификация, УДК, обработка научных текстов, машинное*

обучение, иерархическая классификация, многозначная классификация, математические тексты, цифровые библиотеки, векторизация текста.

ВВЕДЕНИЕ

Универсальная десятичная классификация (УДК) применяется для тематического индексирования научных публикаций, однако ручное присвоение кодов трудоемко и подвержено субъективности, что затрудняет масштабирование на большие массивы текстов. Поэтому актуальна разработка методов автоматического определения УДК по содержанию документа на основе подходов обработки естественного языка и машинного обучения.

Для электронных библиотек и научных архивов задача автоматического индексирования имеет не только исследовательское, но и прикладное значение: от качества тематической классификации документа зависят полнота тематического поиска, корректность навигации по коллекциям и возможность последующего анализа структуры фонда. Ранние работы по автоматической классификации в библиотечно-информационных системах показали, что центральной проблемой остается согласование текстового содержания документа с формальной системой знаний, принятой в конкретной предметной области [1].

Для математических публикаций эта задача осложняется спецификой материала. Помимо общеязыковой и научной лексики, тексты содержат формулы, символические обозначения и устойчивые терминологические сочетания, характерные для отдельных разделов математики. Поэтому границы между классами зависят как от выбора признакового пространства, так и от полноты текстового представления: одна и та же статья может сочетать общетеоретическую лексику и узкоспециальные термы, указывающие на более точный код УДК.

Дополнительную сложность создают иерархическая структура УДК и близость отдельных предметных областей; междисциплинарные тексты могут соответствовать нескольким кодам, а границы между классами часто размыты. Цель настоящей работы – сравнить модели автоматического присвое-

ния кодов УДК для математических публикаций и проанализировать структуру ошибок, включая наиболее информативные термы для различных кодов и те области УДК, которые труднее всего различать между собой.

ИССЛЕДОВАНИЯ, БЛИЗКИЕ ПО ТЕМАТИКЕ

В работах по автоматической классификации кодов УДК одним из наиболее популярных является подход, основанный на алгоритмах машинного обучения. Например, в статье [2] использованы алгоритмы машинного обучения и обработки текста, включая TF-IDF, косинусное сходство, наивный байесовский классификатор и многослойный перцептрон. В [3] исследована эффективность различных архитектур искусственных нейронных сетей для автоматической классификации научных статей по УДК и отмечены возможные области практического применения таких систем. В [4] для определения кода УДК применены алгоритмы SVM и *k*-ближайших соседей.

В более поздних работах все чаще рассматривались нейросетевые и рекомендательные подходы к классификации документов по УДК. В [5] исследовано применение предобученной модели BERT для полуавтоматической предметной идентификации документов и построения рекомендательной системы присвоения кодов УДК. В [6] предложен гибридный подход на основе рекомендательной системы; для его оценки использованы метрики качества ранжирования NDCG, MRR и MAP, а среди наиболее результативных конфигураций были варианты, сочетающие BM25, BERT и дополнительные этапы переупорядочивания рекомендаций. В работе [7] рассмотрена задача классификации научных статей с использованием глубоких нейронных сетей с учетом иерархической структуры УДК. Авторы отмечают, что в некоторых случаях ошибки классификации вызваны некорректно проставленными кодами. В работе [8] рассмотрено использование больших языковых моделей (Large Language Model, LLM) в роли рекомендательной системы для подбора кодов УДК и сравнены различные LLM, что дополняет классические подходы альтернативной парадигмой автоматизации индексирования.

Отдельное направление связано не только с выбором модели, но и с явным учетом иерархической природы меток. В обзоре [9] подчеркнуто, что для иерархической классификации недостаточно плоских метрик: при оценке

качества важно учитывать совпадение предсказанных и истинных меток вместе с их предками в иерархии. Обзор по иерархической классификации текстов [10] и обзор по иерархической многозначной классификации [11] показывают, что методы оценки, декодирования и выбора порогов должны учитывать структуру целевого пространства меток. Для УДК это особенно важно, поскольку иерархия классов задает смысловую близость соседних подклассов и допускает присвоение одному документу нескольких кодов.

Более широкий контекст задают обзоры по автоматической классификации текстов [12–14], в которых систематизируются основные типы текстовых представлений и моделей – от разреженных векторных схем и распределенных представлений до глубоких нейросетевых архитектур. В этих работах показано, что сравнительная эффективность методов определяется не только архитектурой модели, но и свойствами корпуса, включая объем и структуру обучающей выборки, баланс классов и языковые либо доменные особенности данных.

В развитии подходов к представлению текста в задачах автоматической обработки и классификации прослеживается переход от распределенных векторных представлений слов [15] к контекстным моделям на основе трансформеров [16] и далее к специализированным энкодерам, обученным с учетом специфики научного дискурса [17]. Для корпуса математических статей такой переход представляет практический интерес: специализированные модели потенциально лучше учитывают особенности научного дискурса, тогда как лексико-частотные и иные разреженные признаки могут оставаться полезными при различении близких тематик. Именно поэтому в настоящей работе сопоставляются методы, основанные на различных типах текстового представления, но оцениваемые в едином экспериментальном контуре.

Таким образом, анализ литературы показал, что задача автоматического присвоения кодов УДК находится на пересечении нескольких исследовательских направлений: библиотечно-информационного индексирования, иерархической и многозначной классификации, а также обработки научных текстов. Это делает сопоставление методов на математическом корпусе статей на русском языке самостоятельной и практически значимой задачей.

МЕТОД

Корпус формировался автоматически на основе статей с ресурса Math-Net.Ru: для выбранных журналов были отобраны русскоязычные статьи, опубликованные не ранее 2000 г.; для каждой статьи сохранялись PDF-файл полного текста и метаданные (год, заголовок, аннотация, авторы). Итоговый объем корпуса составил 4194 статьи.

Из файлов в формате PDF извлекались коды УДК, год, основной текст и формулы; далее выполнялась очистка от артефактов верстки (служебные блоки, колонтитулы, шапки/подвалы). Для методов TF-IDF и Word2Vec дополнительно выполнялись лемматизация и удаление стоп-слов.

На этапе подготовки корпуса существенным было приведение кодов УДК к сопоставимому виду. В рамках настоящей работы анализ ограничивается глубиной до одного знака после первой точки, что позволяет, с одной стороны, сохранить различимость основных предметных ветвей внутри математической области, а с другой – снизить чувствительность модели к избыточной дробности и единичным вариантам разметки. Такое решение особенно важно для сравнительного исследования, в котором требуется сопоставить модели на едином и достаточно устойчивом уровне детализации.

Для проверки обобщающей способности использовалось временное разбиение: обучение проводилось на статьях до 2020 г., тестирование – на статьях, опубликованных с 2021 г. Временное разбиение выбрано намеренно, поскольку в прикладной системе автоматической категоризации модель чаще применяется к новым публикациям. Такой протокол снижает риск завышенной оценки качества за счет тематического и стилистического сходства текстов одного периода, а также позволяет оценить устойчивость признаков и моделей к возможному смещению распределения корпуса во времени.

Сравнивались четыре типа текстовых представлений: TF-IDF, Word2Vec с усреднением векторов слов с TF-IDF-весами (далее W2V), а также энкодерные представления SciRus-tiny и SciRus-tiny3.5 (mlsa-iai-msu-lab). Для моделей семейства SciRus-tiny текст статьи преобразовывался в последовательность фрагментов, для каждого из которых вычислялось векторное представление; затем эмбеддинги агрегировались во взвешенное среднее и нормализовывались, образуя единый вектор документа. В конфигурации SciRus-

tiny3.5+LogReg(abstract) на вход модели подавались заголовок и аннотация статьи, а в конфигурации SciRus-tiny3.5+LogReg (fulltext) – заголовок и очищенный полный текст; схема кодирования и агрегации эмбеддингов в обоих случаях оставалась одинаковой. В качестве классификаторов использовались логистическая регрессия, Complement Naive Bayes (CNB) и градиентный бустинг на основе CatBoost [18]. Гиперпараметры всех конфигураций подбирались в едином контуре с помощью байесовской оптимизации (Optuna [19]) по кросс-валидации внутри обучающего периода; для каждой конфигурации выполнялось по 100 итераций поиска. В многозначной постановке после обучения для каждого класса дополнительно подбирались индивидуальные пороги бинаризации на валидационной выборке.

Выбор именно этих моделей обусловлен стремлением сопоставить методы, различающиеся как по типу представления текста, так и по способу принятия решения. Логистическая регрессия задает интерпретируемую линейную границу и естественно сочетается с разреженными признаками. Complement Naive Bayes включен в сравнение как сильный базовый метод для высокоразмерных разреженных текстовых представлений; кроме того, он нередко оказывается устойчивым при дисбалансе классов, что особенно важно для задачи присвоения кодов УДК. Градиентный бустинг, в свою очередь, позволяет проверить, дает ли нелинейное моделирование дополнительный выигрыш на тех же входных данных. Благодаря этому сравнение оказывается содержательным не только по итоговым метрикам, но и с точки зрения того, какие свойства корпуса и представления текста оказываются критичными для качества классификации.

Особенность кодов УДК состоит в их иерархичности: коды отражают как общую область, так и более узкую тематику. Поэтому при оценке качества и подборе гиперпараметров учитывается не только точное совпадение класса, но и близость предсказания к истинной ветви УДК.

Разделение на single-label и multi-label постановки отражает два различных прикладных сценария. В первом случае система должна выбрать основной, наиболее представительный код, что ближе к задаче первичной каталогизации. Во втором случае модель должна воспроизвести весь набор темати-

ческих индексов, что ближе к задаче поддержки экспертной разметки и проверки уже существующей классификации. Сравнение этих постановок на одном корпусе по-зволяет оценить, насколько одни и те же признаки по-разному работают в режимах выбора одного класса и выбора набора взаимосвязанных классов.

В постановке с единственной целевой меткой (single-label) оптимизируется критерий

$$S = 0.7 \cdot \mathrm{F1}_{\mathrm{macro}} + 0.3 \cdot \mathrm{F1}_{\mathrm{hier}}.$$

Здесь $\mathrm{F1}_{\mathrm{macro}}$ – макроусредненная F1-мера, а $\mathrm{F1}_{\mathrm{hier}}$ вводится так:

$$L(y) = \{y_1, y_2, \ldots, y_n\}, \qquad L(\hat{y}) = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}.$$

$$P = \frac{|L(y) \cap L(\hat{y})|}{|L(\hat{y})|}, \qquad R = \frac{|L(y) \cap L(\hat{y})|}{|L(y)|}.$$

$$F1_{\mathrm{hier}}(y, \hat{y}) = \frac{2PR}{(P + R)},$$

где $y_1, \ldots, y_n$ – уровни рассматриваемой детализации. Мы используем два уровня для этой метрики: y_1 – код до первой точки, y_2 – код с полной рассматриваемой глубиной (до первой цифры после точки).

В постановке с несколькими метками (multi-label) оптимизируется критерий

$$S = 0.5 \cdot \mathrm{F1}_{\mathrm{micro}} + 0.5 \cdot \mathrm{F1}_{\mathrm{macro}}.$$

Здесь $\mathrm{F1}_{\mathrm{micro}}$ – микроусредненная F1-мера, вычисляемая по агрегированным значениям ошибок и верных результатов по всем меткам, поэтому она преимущественно характеризует качество на частотных классах. Вероятности по классам бинаризовались с использованием индивидуальных порогов, которые для каждого класса подбирались на валидации путем максимизации F1 на заданной сетке значений.

Использование индивидуальных порогов принципиально важно для многозначной постановки, поскольку частота классов и характер их совместной встречаемости различаются. Для устойчивых и частых кодов допустим более низкий порог, если это повышает полноту, тогда как для семантически близких ветвей требуется более осторожная бинаризация, уменьшающая

число ложных срабатываний. Таким образом, подбор порогов становится частью общей адаптации модели к структуре корпуса.

Также рассчитывались метрики: доля правильных ответов (accuracy, acc.), метрики на верхнем уровне – до первой точки, 2-го уровня детализации (top-level) и другие стандартные метрики.

ЭКСПЕРИМЕНТЫ

Экспериментальная часть была построена так, чтобы сравнение моделей опиралось не на одну сводную метрику, а на несколько показателей качества. Для однозначной постановки важны прежде всего устойчивость на редких классах, качество вероятностных оценок и способность модели выводить верные коды в верхние позиции ранжирования. Для многозначной постановки, помимо F1-мер, учитывались показатели качества ранжирования кодов и степень совпадения полных наборов меток на уровне документа.

В однозначной постановке (предсказание основного кода) сравнивались семь моделей: TF-IDF + LogReg, TF-IDF + CatBoost, W2V (усреднение Word2Vec с TF-IDF-весами) + LogReg, SciRus-tiny + LogReg, SciRus-tiny3.5 + LogReg(abstract), SciRus-tiny3.5 + LogReg(fulltext) и TF-IDF + CNB.

Табл. 1. Метрики моделей в задаче предсказания первого кода УДК.

Модель	Acc.	Balanced Acc.	Macro-F1	Weighted F1	Hier-F1	LogLoss	Acc.@3	Top-level Acc.	Top-level Macro-F1
TF-IDF+LogReg	0.857	0.702	0.694	0.855	0.889	0.564	0.970	0.920	0.860
TF-IDF+CatBoost	0.828	0.593	0.638	0.820	0.866	0.597	0.964	0.904	0.781
W2V+LogReg	0.763	0.620	0.595	0.770	0.807	1.148	0.936	0.851	0.755
SciRus-tiny+LogReg	0.705	0.486	0.506	0.699	0.751	1.037	0.928	0.797	0.661
SciRus-tiny3.5+ LogReg(abstract)	0.778	0.557	0.598	0.768	0.818	0.722	0.939	0.859	0.713
SciRus-tiny3.5+ LogReg(fulltext)	0.791	0.559	0.597	0.785	0.830	0.688	0.947	0.867	0.743
TF-IDF+CNB	0.826	0.643	0.629	0.825	0.856	2.708	0.917	0.886	0.807

Наиболее устойчивые результаты на рассматриваемом уровне детализации показала модель TF-IDF + LogReg (см. табл. 1): она обеспечивает наилучший баланс между общей точностью, качеством на редких классах и ранжирующими метриками. Модель TF-IDF + CatBoost близка по общей точности, но сильнее чувствительна к дисбалансу классов. Конфигурация W2V + LogReg уступает на детальном уровне, что, вероятно, связано со сглаживанием различий между близкими терминами при усреднении эмбеддингов. Использование SciRus-tiny без дополнительной адаптации также не дало преимущества. Конфигурация TF-IDF+CNB показала результаты, близкие к TF-IDF + LogReg. Обе версии SciRus-tiny3.5 улучшили показатели относительно исходной SciRus-tiny + LogReg; при этом вариант с полным текстом оказался сильнее варианта с аннотацией.

Таким образом, в однозначной постановке TF-IDF + LogReg демонстрирует не только высокую точность, но и наиболее стабильное качество по совокупности метрик, что делает эту модель перспективной для практических сценариев автоматического подбора основного кода УДК.

Анализ ошибок показал, что труднее всего разделяются близкие подклассы внутри одной ветви УДК; наиболее заметная путаница наблюдается

для пары 517.9 ↔ 517.5. Ошибки между разными ветвями (например, между 517.* и 519.*) встречаются реже и обычно связаны с пересечением терминологии в смежных темах. Характерные термы (табл. 2) подтверждают тематичность признаков: 510.5 соответствует теории алгоритмов и вычислимых функций, 510.6 – математической логике, 512.5 – общей алгебре. Визуализация t-SNE [20] для модели W2V (рис. 1) показывает разделимость на верхнем уровне и частичное смешение на внутреннем, что согласуется с профилем ошибок.

Табл. 2. Наиболее информативные термы TF-IDF для трех кодов УДК по коэффициентам логистической регрессии.

Код	Топ 1 терм	Топ 2 терм	Топ 3 терм	Топ 4 терм	Топ 5 терм	Топ 6 терм
510.5	вычислимый	нумерация	шаг	перечислимый	вычислимость	конструкция
510.6	кортеж	логика	пропозициональный	полигон	язык	категоричный
512.5	подгруппа	группа	идеал	алгебра	кольцо	изоморфный

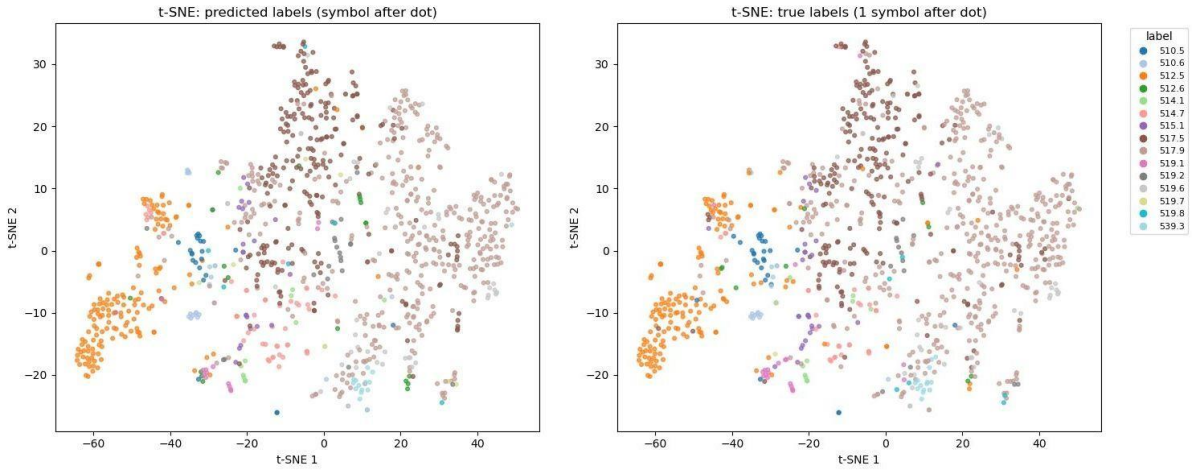


Рис. 1. Визуализация истинных и предсказанных меток кодов УДК с помощью
t-SNE для модели W2V + LogReg.

В свою очередь, такой характер ошибок объясняется иерархической природой задачи. Для прикладной рекомендательной системы смешение

внутри одной тематической ветви обычно менее критично, чем переход в совершенно другую область, именно поэтому наряду с точными метриками в работе анализируются показатели верхнего уровня и иерархическая F1-мера.

В многозначной постановке сравнение проводилось на том же наборе моделей. Вероятности по классам переводились в бинарные решения с использованием индивидуальных порогов, подобранных на валидации для каждого кода. По совокупности метрик наилучшие результаты вновь показала модель

TF-IDF + LogReg: она превосходит альтернативы как по качеству бинарного решения, так и по ранжирующим метрикам. TF-IDF + CNB выступает ближайшей альтернативой, тогда как обе конфигурации SciRus-tiny3.5 улучшают результаты относительно SciRus-tiny+LogReg. При этом вариант с полным текстом устойчиво превосходит вариант, основанный только на заголовке и аннотации.

Распределение числа меток (рис. 2) показывает, что большинство статей имеют один-два кода, а предсказания лучшей модели в целом воспроизводят этот профиль.

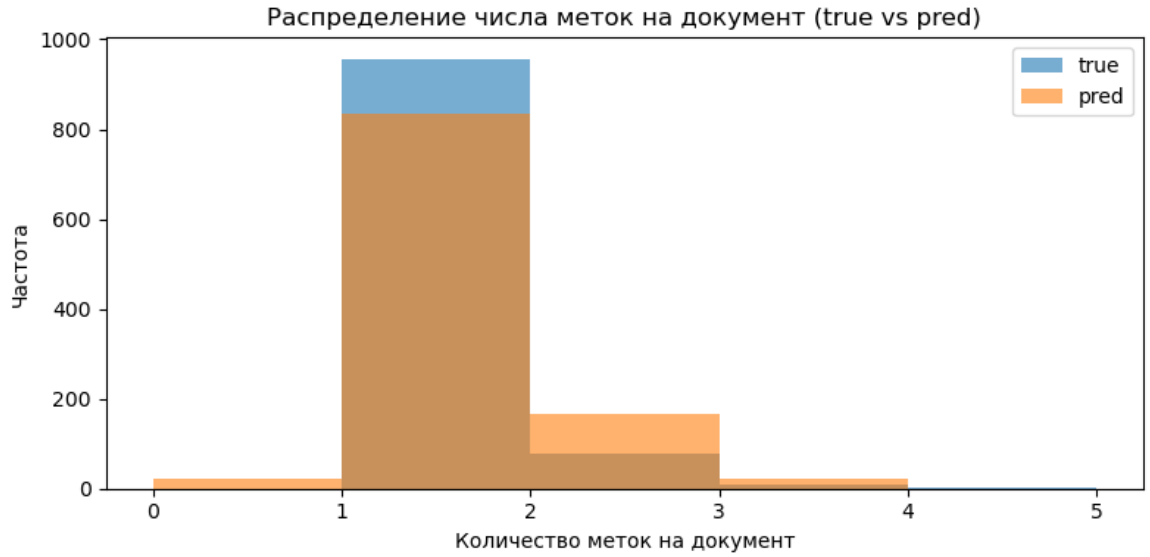


Рис. 2. Распределение числа кодов на документ для модели TF-IDF+LogReg.

Подобранные пороги зависят от частоты и «конкурентности» класса: для устойчивых частых кодов порог может быть ниже, тогда как для семантически близких ветвей он повышается, что уменьшает число ложных срабатываний. Hamming для SciRus-tiny + LogReg оказался ниже остальных моделей из-за того, что модель в целом предсказывала меньше меток и чаще попадала в ответ «такого кода нет». В итоге модель TF-IDF + LogReg превосходит другие рассматриваемые модели и по качеству классификации, и по ранжирующим метрикам (LRAP, mAP), то есть лучше упорядочивает коды (см. табл. 3).

Табл. 3. Метрики моделей в задаче предсказания всех кодов УДК статей.

Модель	Micro-F1	Macro-F1	Samples F1	Hamming	Subset Acc.	LRAP	mAP-micro	mAP-macro	Top-level micro-F1	Top-level macro-F1	Top-level Subset Acc.
TF-IDF + LogReg	0.812	0.646	0.828	0.025	0.717	0.916	0.887	0.727	0.885	0.808	0.820
W2V + LogReg	0.754	0.529	0.756	0.031	0.648	0.871	0.802	0.588	0.833	0.729	0.743
TF-IDF + CatBoost	0.796	0.569	0.796	0.027	0.685	0.891	0.840	0.630	0.871	0.758	0.801
SciRus-tiny + LogReg	0.659	0.149	0.641	0.014	0.531	0.807	0.672	0.302	0.743	0.318	0.640
SciRus-tiny3.5 + LogReg (abstract)	0.724	0.506	0.727	0.035	0.603	0.855	0.797	0.534	0.806	0.621	0.715
SciRus-tiny3.5 + LogReg (fulltext)	0.736	0.544	0.732	0.035	0.599	0.872	0.806	0.592	0.808	0.668	0.694
TF-IDF + CNB	0.790	0.621	0.787	0.027	0.673	0.909	0.848	0.665	0.851	0.776	0.759

Содержательно различие между моделями в multi-label постановке можно интерпретировать так: при множественном присвоении кодов особенно важна способность модели удерживать несколько близких тематических сигналов одновременно.

Разреженные признаки TF-IDF в этих условиях сохраняют различимость частотных и редких терминов, тогда как усреднение плотных векторов сильнее сглаживает различия между соседними тематическими зонами. Это согласуется и с тем, что top-level показатели заметно выше: большая часть ошибок приходится не на выбор неверной верхней ветви, а на выбор соседнего подкласса внутри уже верно определенного направления.

ЗАКЛЮЧЕНИЕ

Исследованы подходы к автоматическому присвоению кодов УДК математическим публикациям на русском языке на корпусе из 4194 статей, собранном на основе ресурса Math-Net.Ru. Оценка проведена на временном разбиении, что позволило более реалистично оценить обобщающие способности моделей.
Задача рассматривалась в однозначной и многозначной постановках на уровне одного знака после первой точки с дополнительным учетом иерархической структуры УДК.

Результаты проведенных экспериментов показали, что при выбранной детализации наиболее стабильные результаты в обеих постановках обеспечивает модель TF-IDF + LogReg. Ее преимущество проявляется не только в основных метриках классификации, но и в показателях, связанных с ранжированием кандидатов, что особенно важно для полуавтоматических сценариев библиотечного индексирования. Конфигурация TF-IDF + CNB выступает сильной альтернативой и дает близкие результаты. Переход от SciRus-tiny к SciRus-tiny3.5 улучшает качество семантических конфигураций, однако даже лучший вариант с полным текстом уступает TF-IDF-базовым моделям на исследуемом корпусе при выбранной агрегации.

С практической точки зрения полученные результаты позволяют рассматривать разработанный подход как основу для нескольких сценариев внедрения: автоматической первичной рубрикации новых поступлений, рекомендаций кодов автору или редактору при подготовке публикации, а также проверки уже существующей разметки в ретроспективных коллекциях. По-

следний сценарий особенно важен для цифровых библиотек, поскольку позволяет использовать модель не только как инструмент предсказания, но и как средство контроля качества метаданных.

В настоящей работе, несмотря на извлечение формульных выражений при обработке документов, модели использовали только текстовые признаки, поэтому вклад формул в качество классификации не оценивался. Дальнейшая работа предполагает включение формульных признаков и оценку их эффекта, поскольку нотация и типовые формульные выражения часто являются предметно-специфичными маркерами разделов математики.

Кроме того, перспективы дальнейших исследований включают расширение коллекции, построение собственных языковых моделей, специфичных для рассматриваемой предметной области, расширение набора методов и ансамблей, более полное сравнение различных источников текста (аннотации и полного текста), а также более глубокий учет иерархии УДК, включая иерархические функции потерь и иерархическое декодирование предсказаний. Эти направления станут продолжением настоящей работы и позволяют перейти от сравнительного исследования к более прикладным библиотечным системам поддержки классификации.

СПИСОК ЛИТЕРАТУРЫ

1. *Tóth E.* Innovative Solutions in Automatic Classification: A Brief Summary // Libri. 2002. Vol. 52, No. 1. P. 48–53. https://doi.org/10.1515/LIBR.2002.48

2. *Romanov A., Lomotin K., Kozlova E.* Automatization of Scientific Articles Classification According to Universal Decimal Classifier // Supplementary Proceedings of the Sixth International Conference on Analysis of Images, Social Networks and Texts (AIST 2017). CEUR Workshop Proceedings. 2017. Vol. 1975. P. 122–133.

3. *Romanov A.Yu., Lomotin K.E., Kozlova E.S., Kolesnichenko A.L.* Research of neural networks application efficiency in automatic scientific articles classification according to UDC // Proceedings of the 2016 International Siberian Conference on Control and Communications (SIBCON 2016), Moscow, Russia, 12–14 May 2016. IEEE, 2016. P. 612–616. https://doi.org/10.1109/SIBCON.2016.7491783

4. *Kragelj M., Kljajić Borštnar M.* Automatic classification of older electronic texts into the Universal Decimal Classification-UDC // Journal of Documentation. 2021. Vol.77, No. 3. P. 755–776. https://doi.org/10.1108/JD-06-2020-0092

5. *Roy A., Ghosh S.* Automated Subject Identification using the Universal Decimal Classification: The ANN Approach // SRELS Journal of Information and Knowledge. 2023. Vol. 60, No. 2. P. 69–76. https://doi.org/10.17821/srels/2023/v60i2/170963

6. *Borovič M., Ojsteršek M., Strnad M.* A Hybrid Approach to Recommending Universal Decimal Classification Codes for Cataloguing in Slovenian Digital Libraries // IEEE Access. 2022. Vol. 10. P. 85595–85605. https://doi.org/10.1109/ACCESS.2022.3198706

7. *Мамедов В.Ю., Ковалевский Д.А., Морозов Д.А., Столяров С.С., Оспичев С.С.* Иерархическая классификация научных статей при помощи глубокого обучения (на примере иерархии УДК) // Моделирование и анализ информационных систем. 2025. Т. 32. № 1. С. 80–94. https://doi.org/10.18255/1818-1015-2025-1-80-94

8. *Borovič M., Tomovski E., Li Dobnik T., Majninger S.* Evaluating Proprietary and Open-Weight Large Language Models as Universal Decimal Classification Recommender Systems // Applied Sciences. 2025. Vol. 15. No. 14. Art. 7666. https://doi.org/10.3390/app15147666

9. *Silla C.N. Jr., Freitas A.A.* A Survey of Hierarchical Classification across Different Application Domains // Data Mining and Knowledge Discovery. 2011. Vol. 22, No. 1–2. P. 31–72. https://doi.org/10.1007/s10618-010-0175-9

10. *Zangari A., Marcuzzo M., Rizzo M., Giudice L., Albarelli A., Gasparetto A.* Hierarchical Text Classification and Its Foundations: A Review of Current Research // Electronics. 2024. Vol. 13, No. 7. Art. 1199. https://doi.org/10.3390/electronics13071199

11. *Liu R., Liang W., Luo W., Song Y., Zhang H., Xu R., Li Y., Liu M.* Recent Advances in Hierarchical Multi-label Text Classification: A Survey. // 2023. arXiv:2307.16265. https://doi.org/10.48550/arXiv.2307.16265

12. *Kowsari K., Jafari Meimandi K., Heidarysafa M., Mendu S., Barnes L.E., Brown D.E.* Text Classification Algorithms: A Survey // Information. 2019. Vol. 10, No. 4. Art. 150. https://doi.org/10.3390/info10040150

13. *Li Q., Peng H., Li J., Xia C., Yang R., Sun L., Yu P.S., He L.* A Survey on Text Classification: From Traditional to Deep Learning // ACM Transactions on Intelligent Systems and Technology. 2022. Vol. 13, No. 2. Art. 31. P. 1–41. https://doi.org/10.1145/3495162

14. *Mirończuk M.M., Protasiewicz J.* A Recent Overview of the State-of-the-Art Elements of Text Classification // Expert Systems with Applications. 2018. Vol. 106. P. 36–54. https://doi.org/10.1016/j.eswa.2018.03.058

15. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space // 2013. arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

16. *Devlin J., Chang M.-W., Lee K., Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of NAACL-HLT 2019. Minneapolis, Minnesota, 2019. P. 4171–4186. https://doi.org/10.18653/v1/N19-1423

17. *Герасименко Н.А., Ватолин А., Янина А., Воронцов К.В.* SciRus: легкий и мощный мультиязычный энкодер для научных текстов // Доклады Российской академии наук. Математика, информатика, процессы управления. 2024. Т. 520, № 2. С. 216–227. https://doi.org/10.1134/S1064562424602178

18. *Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., Gulin A.* CatBoost: unbiased boosting with categorical features // Advances in Neural Information Processing Systems. 2018. Vol. 31. P. 6638–6648. https://doi.org/10.48550/arXiv.1706.09516

19. *Akiba T., Sano S., Yanase T., Ohta T., Koyama M.* Optuna: A Next-generation Hyperparameter Optimization Framework // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019. P. 2623–2631. https://doi.org/10.1145/3292500.3330701

20. *van der Maaten L., Hinton G.* Visualizing Data using t-SNE // Journal of Machine Learning Research. 2008. Vol. 9, No. 86. P. 2579–2605.

METHODS FOR AUTOMATIC ASSIGNMENT OF UDC CODES TO MATHEMATICAL ARTICLES: AN EVALUATION OF CLASSICAL AND NEURAL APPROACHES

B. T. Gizatullin[1] [0009-0000-6251-9260], O. A. Nevzorova[2] [0000-0001-8116-9446]

[1, 2]*Kazan (Volga Region) Federal University, Kazan, Russia*

[1]gizat.blt@gmail.com, [2]onevzoro@gmail.com

Abstract

Universal Decimal Classification (UDC) is a hierarchical indexing system in which a publication may be assigned one or several codes. Manual UDC indexing is labor-intensive and often inconsistent. This paper addresses the automatic assignment of UDC codes to Russian-language mathematical research articles. The aim is to compare combinations of text representations and classification models on a unified corpus and to identify the most effective configurations. A corpus of 4194 articles was collected from Math-Net.Ru, including full texts, abstracts, metadata, and UDC codes. The preprocessing pipeline comprised PDF text extraction, removal of layout artifacts, and normalization of UDC labels. We compared TF-IDF, Word2Vec, SciRus-tiny, and SciRus-tiny3.5 representations combined with logistic regression, Complement Naive Bayes (CNB), and CatBoost. In both the single-label and multi-label settings, the best performance was achieved by TF-IDF + LogReg, while TF-IDF + CNB showed closely competitive results. The proposed approach can be used in automatic subject indexing systems for digital libraries and scientific archives, in UDC recommendation tools for authors and editors, and in metadata quality control workflows.

Keywords: *automatic classification, Universal Decimal Classification, UDC, scientific text processing, machine learning, hierarchical classification, multi-label classification, mathematical texts, digital libraries, text vectorization.*

REFERENCES

1. *Tóth E.* Innovative Solutions in Automatic Classification: A Brief Summary // Libri. 2002. Vol. 52, No. 1. P. 48–53. https://doi.org/10.1515/LIBR.2002.48

2. *Romanov A., Lomotin K., Kozlova E.* Automatization of Scientific Articles Classification According to Universal Decimal Classifier // Supplementary Proceedings of the Sixth International Conference on Analysis of Images, Social Networks and Texts (AIST 2017). CEUR Workshop Proceedings. 2017. Vol. 1975. P. 122–133.

3. *Romanov A.Yu., Lomotin K.E., Kozlova E.S., Kolesnichenko A.L*. Research of neural networks application efficiency in automatic scientific articles classification according to UDC // Proceedings of the 2016 International Siberian Conference on Control and Communications (SIBCON 2016), Moscow, Russia, 12–14 May 2016. IEEE, 2016. P. 612–616. https://doi.org/10.1109/SIBCON.2016.7491783

4. *Kragelj M., Kljajić Borštnar M*. Automatic classification of older electronic texts into the Universal Decimal Classification-UDC // Journal of Documentation. 2021. Vol. 77, No. 3. P. 755–776. https://doi.org/10.1108/JD-06-2020-0092

5. *Roy A., Ghosh S.* Automated Subject Identification using the Universal Decimal Classification: The ANN Approach // SRELS Journal of Information and Knowledge. 2023. Vol. 60. No. 2. P. 69-76. https://doi.org/10.17821/srels/2023/v60i2/170963

6. *Borovič M., Ojsteršek M., Strnad M.* A Hybrid Approach to Recommending Universal Decimal Classification Codes for Cataloguing in Slovenian Digital Libraries // IEEE Access. 2022. Vol. 10, P. 85595–85605. https://doi.org/10.1109/ACCESS.2022.3198706

7. *Mamedov V., Kovalevsky D., Morozov D., Stolyarov S., Ospichev S*. Hierarchical classification of scientific articles using deep learning (using the UDC hierarchy as an example) // Modeling and Analysis of Information Systems. 2025. Vol. 32, No. 1. P. 80–94. https://doi.org/10.18255/1818-1015-2025-1-80-94

8. *Borovič M., Tomovski E., Li Dobnik T., Majninger S.* Evaluating Proprietary and Open-Weight Large Language Models as Universal Decimal Classification Recommender Systems // Applied Sciences. 2025. Vol. 15, No. 14. Art. 7666. https://doi.org/10.3390/app15147666

9. *Silla C.N. Jr., Freitas A.A.* A Survey of Hierarchical Classification across Different Application Domains // Data Mining and Knowledge Discovery. 2011. Vol. 22, No. 1–2. P. 31–72. https://doi.org/10.1007/s10618-010-0175-9

10. *Zangari A., Marcuzzo M., Rizzo M., Giudice L., Albarelli A., Gasparetto A*. Hierarchical Text Classification and Its Foundations: A Review of Current Research // Electronics. 2024. Vol. 13, No. 7. Art. 1199. https://doi.org/10.3390/electronics13071199

11. *Liu R., Liang W., Luo W., Song Y., Zhang H., Xu R., Li Y., Liu M.* Recent Advances in Hierarchical Multi-label Text Classification: A Survey // 2023. arXiv:2307.16265. https://doi.org/10.48550/arXiv.2307.16265

12. *Kowsari K., Jafari Meimandi K., Heidarysafa M., Mendu S., Barnes L.E., Brown D.E.* Text Classification Algorithms: A Survey // Information. 2019. Vol. 10, No. 4. Art. 150. https://doi.org/10.3390/info10040150

13. *Li Q., Peng H., Li J., Xia C., Yang R., Sun L., Yu P.S., He L.* A Survey on Text Classification: From Traditional to Deep Learning // ACM Transactions on Intelligent Systems and Technology. 2022. Vol. 13, No. 2. Art. 31. P. 1–41. https://doi.org/10.1145/3495162

14. *Mirończuk M.M., Protasiewicz J.* A Recent Overview of the State-of-the-Art Elements of Text Classification // Expert Systems with Applications. 2018. Vol. 106. P. 36–54. https://doi.org/10.1016/j.eswa.2018.03.058

15. *Mikolov T., Chen K., Corrado G., Dean J.* Efficient Estimation of Word Representations in Vector Space // 2013. arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

16. *Devlin J., Chang M.-W., Lee K., Toutanova K.* BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of NAACL-HLT 2019. Minneapolis, Minnesota, 2019. P. 4171–4186. https://doi.org/10.18653/v1/N19-1423

17. *Gerasimenko N., Vatolin A., Ianina A., Vorontsov K.* SciRus: Tiny and Powerful Multilingual Encoder for Scientific Texts // Doklady Mathematics. 2024. Vol. 110, Suppl. 1. P. S193–S202. https://doi.org/10.1134/S1064562424602178

18. *Prokhorenkova L., Gusev G., Vorobev A., Dorogush A.V., Gulin A.* CatBoost: unbiased boosting with categorical features // Advances in Neural Information Processing Systems. 2018. Vol. 31. P. 6638–6648. https://doi.org/10.48550/arXiv.1706.09516

19. *Akiba T., Sano S., Yanase T., Ohta T., Koyama M.* Optuna: A Next-generation Hyperparameter Optimization Framework // Proceedings of the 25th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019. P. 2623–2631. https://doi.org/10.1145/3292500.3330701

20. *van der Maaten L., Hinton G.* Visualizing Data using t-SNE // Journal of Machine Learning Research. 2008. Vol. 9, No. 86. P. 2579–2605.

СВЕДЕНИЯ ОБ АВТОРАХ

ГИЗАТУЛЛИН Булат Тимурович – магистрант Института математики и механики им. Н. И. Лобачевского Казанского (Приволжского) федерального университета, направление подготовки «Математика и компьютерные науки», профиль «Статистические методы науки о данных».

Bulat Timurovich GIZATULLIN – master's student at the Lobachevsky Institute of Mathematics and Mechanics, Kazan (Volga Region) Federal University, program in Mathematics and Computer Science, track "Statistical Methods in Data Science".

email: gizat.blt@gmail.com

ORCID: 0009-0000-6251-9260

НЕВЗОРОВА Ольга Авенировна – кандидат технических наук, доцент Казанского (Приволжского) федерального университета.

Olga Avenirovna Nevzorova – Candidate of Technical Sciences (PhD equivalent), associate Professor at Kazan (Volga Region) Federal University.

email: onevzoro@gmail.com

ORCID: 0000-0001-8116-9446

Материал поступил в редакцию 14 апреля 2026 года

УДК 004.81

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ОЦЕНКЕ ГРАФОВ ЗНАНИЙ В ДОМЕННОЙ ОБЛАСТИ МАШИНОСТРОИТЕЛЬНЫХ СИСТЕМ ПОЛНОГО ЖИЗНЕННОГО ЦИКЛА

В. В. Гладышев[0009-0003-8900-3469]

Московский физико-технический институт; Центр «Пуск», г. Долгопрудный, Россия

gladyshev.vv@phystech.edu

Аннотация

Работа посвящена проблеме применения онтологического подхода при построении датасета для оценки и сравнения систем обогащения контекста большой языковой модели с использованием графов знаний в доменной области машиностроительных систем полного жизненного цикла. В доменной области сложно получить необходимое количество текстовых данных с формальной логической структурой для формирования оценочного набора без использования сгенерированных синтетических данных. Для исключения внесения искажений и галлюцинаций при формировании оценочного набора предложено оригинальное решение проблемы дефицита данных за счет извлечения онтологии непосредственно из файлов изделий и сборок, соответствующих стандарту STandard for Exchange of Product model data что потенциально позволяет использовать все данные об изделиях как источник для масштабирования оценочных данных. Целью работы стали создание датасета структурированных текстовых данных в доменной области машиностроительных систем полного жизненного цикла, разработка методики оценки и реализация конвейеров обогащения контекста большой языковой модели с применением и без применения графов знаний для анализа вклада систем с извлечением структуры данных в качество генерируемых ответов. Предложен новый источник оценочных данных, разработана новая методика формирования текстовых оценочных данных с сохранением логической

структуры, реализован конвейер для использования сгенерированных оценочных данных. Получены результаты оценки, подтверждающие положительный вклад систем с извлечением структурированных данных в качество генерируемых ответов в доменной области машиностроительных систем полного жизненного цикла.

Ключевые слова: *онтология, датасет, система полного жизненного цикла СПЖЦ/PLM, система автоматизированного проектирования САПР/CAD, большая языковая модель БЯМ/LLM, генерация с обогащением контекста RAG, GraphRAG, STandard for Exchange of Product model data – STEP.*

ВВЕДЕНИЕ

Для построения информационных систем, обеспечивающих использование доменно-ориентированных данных, предназначенных для локального размещения, часто используется подход обогащения контекста (Retrieval-Augmented Generation, RAG) большой языковой модели (БЯМ, Large Language Model, LLM). Стандартный подход обогащения контекста имеет ряд принципиальных ограничений, для преодоления которых применяется генерация графов знаний при помощи БЯМ GraphRAG [1]. Ключевая проблема заключается в том, что основная масса применяемых средств (датасетов, бенчмарков, метрик) оценки качества подхода обогащения контекста БЯМ не предназначена для оценки GraphRAG, так как не учитывает преимущества структурирования данных, а нацелена на оценку качества извлечения отдельных фактов [2]. Особенно остро данная проблема выражена при построении систем в относительно узких доменных областях и, если для медицинского или юридического направлений существует ограниченное количество доменно-ориентированных данных, для домена машиностроительных систем полного жизненного цикла (Product Lifecycle Management – PLM) таких данных найти не удалось. Соответственно, невозможно обосновать (или отвергнуть) необходимость применения более ресурсоемкого GraphRAG-подхода, а также достоверно оценить влияние изменений, вносимых в GraphRAG-систему, на качество ответов [3].

Применение GraphRAG-систем

Традиционный подход RAG решает проблему актуальности знаний БЯМ, извлекая релевантные фрагменты текста (чанки) из внешнего хранилища и передавая их в БЯМ для генерации ответа. Однако этот подход имеет существенный недостаток: он оперирует с изолированными текстовыми фрагментами, не позволяя выполнять обобщение по всему тексту [4]. GraphRAG представляет собой следующий эволюционный шаг, где извлечение происходит из графа знаний [5]. В этом подходе неструктурированный текст сначала преобразуется в структурированное графовое представление: сущности из текста становятся узлами графа, а отношения между ними — ребрами, выполняется многоуровневая кластеризация и обобщение для каждой группы [6]. Это позволяет системе строить ответ на связанных данных, находя цепочки связей между сущностями, которые не упоминаются в одном и том же предложении или документе. Эффективность GraphRAG была продемонстрирована в ряде исследований. В работе [7] сравнивается GraphRAG, построенный на основе экспертно-разработанной онтологической схемы, с базовыми векторными методами RAG. GraphRAG достиг 90%-ной точности в ответах на 20 сложных вопросов в узкой предметной области, в то время как базовый RAG – лишь 60%.

Ключевые стандарты данных в системах управления жизненным циклом продукции

Современные системы управления жизненным циклом продукции (СПЖЦ [8]; Product Lifecycle Management, PLM) в машиностроительной отрасли представляют собой сложные информационные системы, интегрирующие множество модулей: САПР/CAD, CAE, CAM, CAPP, PDM, ERP, MES и др. Данные, накапливаемые в этих системах, имеют специфическую структуру (геометрия деталей, деревья истории построения, составы сборок) и различный формат для каждого модуля. Для решения задачи формирования структурированных оценочных данных для оценки GraphRAG применяют данные в формате ISO 10303 [9] (STandard for Exchange of Product model data, STEP), стандартном для домена машиностроительных PLM-систем. Стандарт ISO 10303 разделяется на

700 базовых стандартов, в его частях 11–18 и 21 описаны язык определения схем данных EXPRESS и STEP-формат. В стандарте также описаны прикладные протоколы (Application Protocols, AP) для представления специфических данных (AP238 для CAM, AP203 и 242 для САПР/CAD, AP 214 для автомобильной промышленности). Национальный институт стандартов и технологий (The National Institute of Standards and Technology, NIST) предоставляет различные инструменты для просмотра и анализа файлов STEP. Схема данных в EXPRESS-формате и соответствующие ей данные конкретной детали или сборки в STEP-формате могут быть преобразованы в виде онтологии в OWL-формате [10]. В NIST разработаны специальные программные средства открытого доступа: OntoSTEP, STEPCode, STP2OWL.

1. МЕТОДЫ

1.1. Создание датасета для оценки GraphRAG в сравнении с тривиальным RAG

Для формирования оценочного набора корректно структурированных данных нами реализована методика, включающая следующие этапы.

Этап 1. Формирование онтологии из инженерных данных STEP

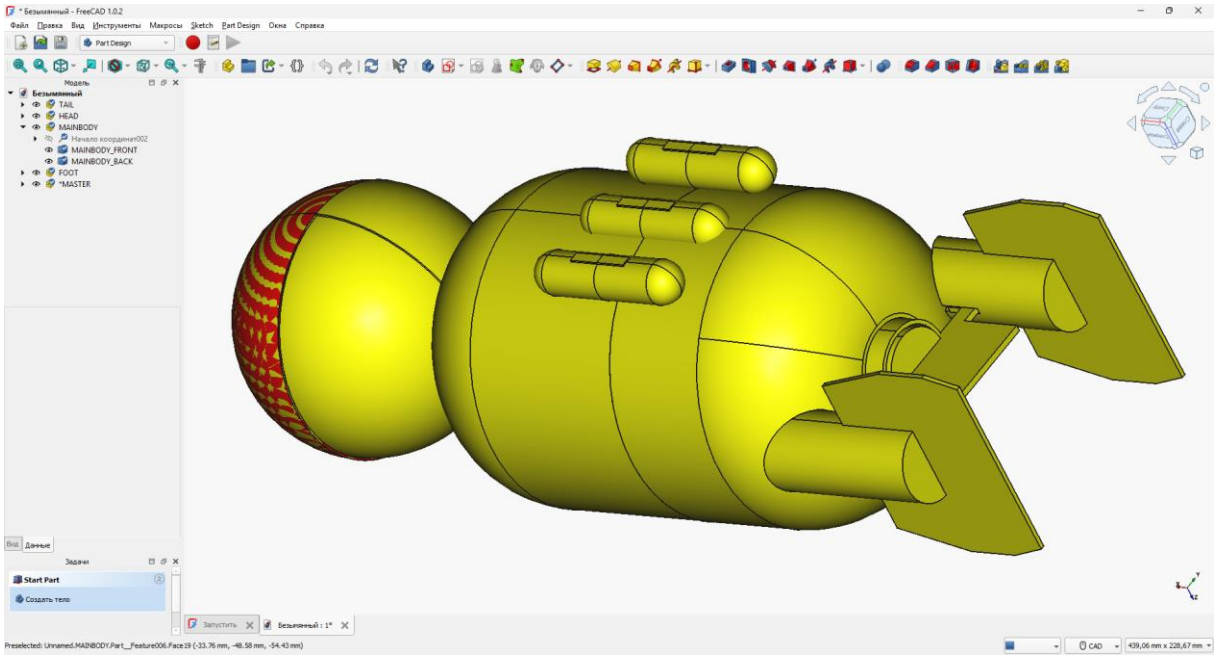


Рис. 1. Сборка s1-c5-214.stp в формате AP214 при просмотре во FreeCAD.

В качестве основы для генерации датасета была использована онтология, автоматически сгенерированная с помощью инструментов Stepcode и STP2OWL. Для извлечения онтологий была использована сборка s1-c5-214 в формате AP214 (STEP-файл с описанием геометрии и сопряжения элементов). Входными данными были файл сборки s1-c5-214.stp в STEP-формате (рис. 1) и логическая схема данных стандарта AP214E3_2010.exp AP214 в EXPRESS-формате [11]. Преобразование выполнялось при помощи программных средств открытого доступа Stepcode и STP2OWL.

На выходе получены онтологии:

– структуры данных s1-c5-214_Schema_DL.owl (Schema – TBox), более 13000 аксиом;

– экземпляра сборки s1-c5-214_Instances_DL.owl (Instances – ABox), около 1000 экз. (рис. 2).

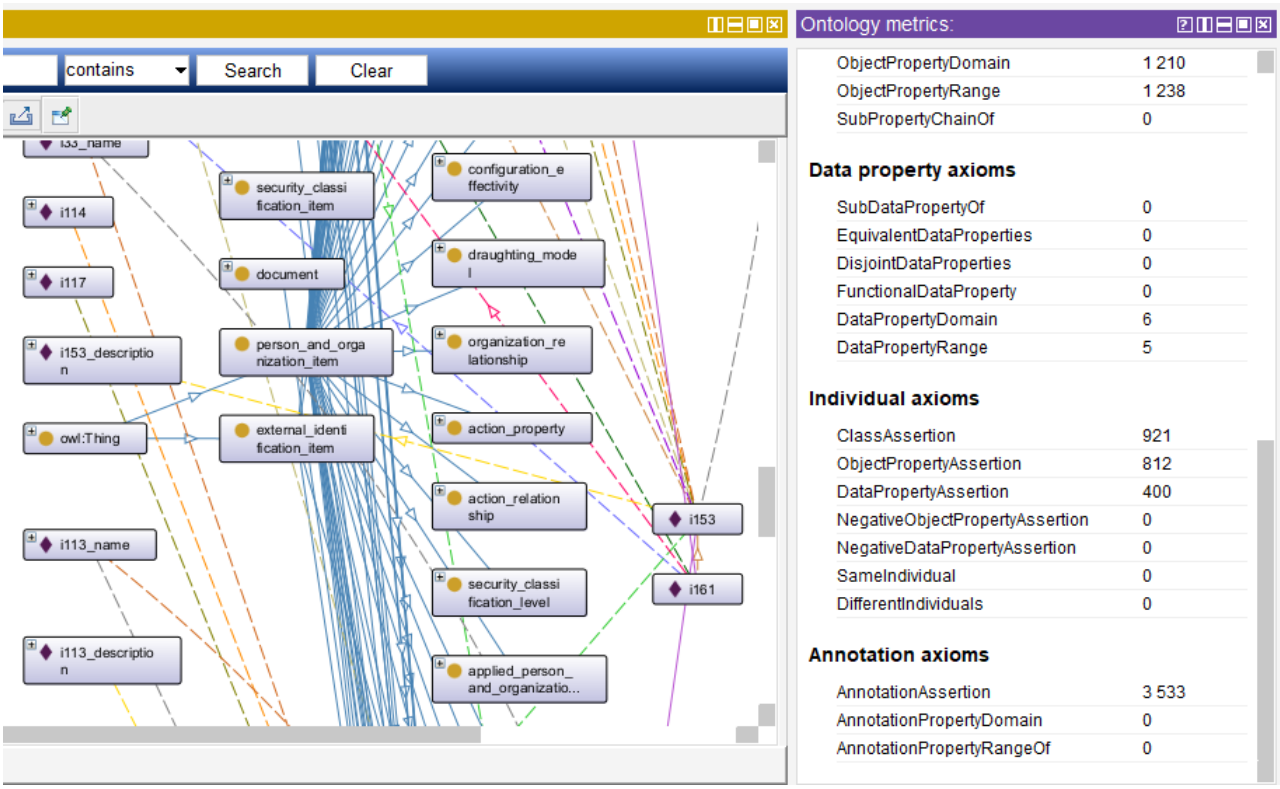


Рис. 2. Связи классов и статистика аксиом экземпляра сборки.

Этап 2. Вербализация онтологии в текстовый датасет

Системы на базе RAG и GraphRAG предназначены для работы с текстом [12]. Поэтому для применения полученной онтологии для построения графа знаний GraphRAG необходимо выполнить преобразование (вербализацию/ verbalization) онтологии в текст. Для сохранения структуры отношений и реализации однозначного преобразования в качестве целевого языка был выбран Attempto Controlled English (ACE) [13], разработанный в Цюрихском университете. Это особое подмножество английского языка, предназначенное для однозначного представления знаний и запросов. Для преобразования использовался OWL Ontology Verbalizer [14]. После преобразования онтологии структуры данных (Schema) получен текстовый файл с аксиомами в ACE-формате. После преобразования онтологии экземпляра сборки (Instances) получен файл с текстами аксиом.

Выбор Attempto Controlled English в качестве целевого языка обусловлен тем, что:

– после преобразования получается интерпретируемый текст – понятный как при чтении человеком, так и корректно интерпретируемый БЯМ [15];

– преобразование сохраняет структурные связи и является формализованным и детерминированным, что исключает вероятностную составляющую при преобразовании с помощью БЯМ;

– преобразование выполняется автоматически и не требует сложной обработки/подготовки;

– преобразование потенциально обратимое.

1.2. Возможные ограничения при формировании датасета

Предлагаемая методика потенциально позволяет произвольно увеличивать размер целевого набора данных при условии разрешения возможного конфликта извлекаемых имен объектов.

Построение датасета на основании онтологий, извлекаемых из схем данных в EXPRESS-формате и STEP-файлов с последующей вербализацией в ACE, имеет одно существенное ограничение в части масштабирования: неполная

поддержка предлагаемого NIST-стандарта на практике. Извлечение ABox выполняется успешно не для всех файлов STEP-формата, что может существенно ограничить базу для формирования целевого набора данных. Кроме того, степень соответствия стандарту в реализациях различных вендоров PLM-систем требует дополнительного исследования.

Этап 3. Анализ интерпретируемости ACE-формата и формирование выборки аксиом

Для анализа интерпретируемости большими языковыми моделями полученного текста в ACE-формате был выполнен ряд экспериментов с увеличением сложности.

1.3. Проверка способности БЯМ выполнять операции с ACE-текстом

Для проверки способности БЯМ интерпретировать ACE-текст – извлекать аксиомы, логически связанные с некоторым объектом, форматировать ответ в ACE-формате – было подготовлено тестовое задание. Произвольным образом был выбран объект с именем i101 и выполнены поиск и отбор аксиом (в отдельный файл), связанных как с самим объектом, так и с его свойствами. Полученная выборка включает 40 аксиом, что позволяет интерпретировать выводы модели и визуально проанализировать их качество.

Данная проверка имитирует ситуацию в информационной системе на базе обогащения контекста, когда фазы излечения и обогащения сформировали качественный (релевантный и полный) контекст.

Проверка выполнялась на трех БЯМ: Anthropic Claude Sonnet 4.5, Qwen 3 Max и OpenAI GPT-5-mini. Первые две модели выбраны, так как используют подход рассуждения и демонстрируют актуальные значения качества в бенчмарках. Модель GPT-5-mini выбрана, поскольку имеет оптимальное соотношение цены и качества и в дальнейшем использовалась для построения графа знаний в библиотеке Microsoft GraphRAG, что потребовало нескольких сотен обращений к модели.

Было выполнено по два запроса к каждой модели. Промпты включали отобранный набор аксиом:

Промпт 1 – команда отобразить набор аксиом, логически связанных с объектом i101: `Answer with Attempto Controlled English. Show the axioms logically related to i101;`

Промпт 2 – команда сделать логические выводы относительно объекта i101 на основании представленных аксиом: `Answer with Attempto Controlled English. Draw logical conclusions based on the axioms regarding i101.`

Проведенные эксперименты продемонстрировали способность БЯМ интерпретировать текст в ACE-формате; извлекать контекстно связанные аксиомы; делать отдельные заключения, исходя из аксиом; формировать ответы с использованием аксиом. Соответственно, датасет, формируемый из ACE-аксиом, применим для оценки информационных систем на базе RAG и GraphRAG. Включение аксиом в ответы БЯМ позволяет реализовать систему оценки с опорой на формальные признаки, выраженные в численном виде [16].

1.4. Формирование выборки аксиом для оценки GraphRAG и тривиального RAG

Специфика реализации фазы извлечения (retrieve) RAG-конвейера заключается в извлечении ограниченного набора документов или чанков для формирования контекста при генерации ответа при помощи БЯМ [17]. В реальных документах информация о каком-либо объекте часто распределена по различным разделам и подразделам. Например, в документации на модуль САПР/CAD описано несколько различных средств построения сечений и также сечения используются как одна из операций при различных построениях [18]. Различные виды сечений и их упоминания распределены по многим разделам документации, что затрудняет формирование контекста при использовании тривиального RAG-подхода, в котором ограничено количество извлекаемых чанков. Соответственно, при построении выборки аксиом выполняется перемешивание для имитации распределения информации по набору документов. Кроме того, было выполнено разделение полученной выборки аксиом на 66 отдельных текстовых документа. Полученный набор файлов с аксиомами использовался для построения библиотекой Microsoft GraphRAG конвейера RAG и соответствующего графа.

Для оценки конвейеров был сформирован набор тестовых вопросов для каждого из 198 объектов (i1 – i198). Тексты вопросов одинаковые: необходимо описать связи объекта с другими объектами. Для каждого вопроса при помощи БЯМ был сгенерирован эталонный ответ с использованием гарантированного полного и достоверного набора аксиом (извлечены поиском по полному набору).

2. ЭКСПЕРИМЕНТЫ

2.1. Построение конвейера RAG

Для сравнения с GraphRAG был реализован классический конвейер RAG. В этом конвейере каждый документ разделяется на чанки. Для каждого чанка векторное представление формируется и затем сохраняется в векторную базу данных. При получении запроса пользователя для него также формируется векторное представление и выполняется ранжирование векторов чанков на основании косинусного расстояния. Тексты пяти наиболее близких чанков передаются БЯМ в качестве контекста запроса в фазе генерации.

Оценка с использованием метрик из библиотеки RAGAS. Для оценки конвейера RAG брались специальные метрики из библиотеки RAGAS [19]. Метрики для оценки конвейера RAG используют подход LLM-as-a-Judge (LLM-based evaluation) — подход в обработке естественного языка, при котором БЯМ используются как автоматические оценщики текстов или других выходных данных моделей. Такой подход требует большого количества обращений к БЯМ и является ресурсоемким, поэтому был выполнен отбор метрик по критерию ресурсоемкости и длительности вычисления на части оценочного набора вопросов. В табл. 1 приведены результаты оценки времени расчета метрик на фрагменте данных.

Табл. 1. Сравнение длительности вычисления.

Метрика	Среднее время	Общее время	Доля, %
Context Precision	58.06	290.3	30.9
Answer Relevancy	35.40	177.0	18.9
Faithfulness	29.55	147.8	15.8
Context Recall	17.19	86.0	9.2
Context Relevance	16.90	84.5	9.0
Response Groundedness	15.62	78.1	8.3
Answer Accuracy	14.87	74.3	7.9

Были отобраны метрики, оптимальные по соотношению значимости и ресурсоемкости.

Answer Relevancy (релевантность ответа) – насколько сгенерированный ответ соответствует исходному вопросу по смыслу, без оценки фактической точности. Штрафует ответы, неполные или содержащие избыточную информацию. Входные данные: question (вопрос), answer (сгенерированный ответ).

Context Recall (полнота контекста) – насколько контекст, извлеченный поисковой системой, содержит всю информацию, необходимую для ответа на вопрос. Метрика требует наличия эталонного (ground truth) ответа. Входные данные: question (вопрос), retrieved context (извлеченный контекст), ground truth answer (эталонный ответ).

Context Relevance (релевантность контекста) – измеряет релевантность вопросу выбранного контекста, помогая улучшить выбор контекста для повышения точности ответа. Входные данные: question (вопрос), retrieved context (извлеченный контекст).

Answer Accuracy (точность ответа) – измеряет соответствие между ответом модели и эталонным значением истинности для данного вопроса. Это делается с помощью двух отдельных запросов “LLM-as-a-Judge”, каждый из которых возвращает оценку (0, 2 или 4). Метрика преобразует эти оценки в шкалу [0,1], а затем вычисляет среднее значение двух оценок, полученных от судей. Более

высокие оценки указывают на то, что ответ модели полностью соответствует эталону. Входные данные: question (вопрос), answer (ответ).

Значения метрик конвейера RAG на наборе из 198 вопросов: Answer Relevancy – 0.359, Context Recall – 0.116, Context Relevance – 0.551, Answer Accuracy – 0.367

2.2. Построение графа знаний с помощью библиотеки GraphRAG

Для проверки применимости полученных данных для построения графов знаний был построен граф знаний при помощи библиотеки Microsoft GraphRAG [1]: по полному набору аксиом об экземпляре сборки и сокращенному набору контекста объекта i101. Для извлечения графа использовалась локальная модель Google Gemma 3 27B. Полный набор аксиом сборки – s1-c5-214. Граф включает 523 узла – извлеченные сущности и 539 связей (рис. 3).

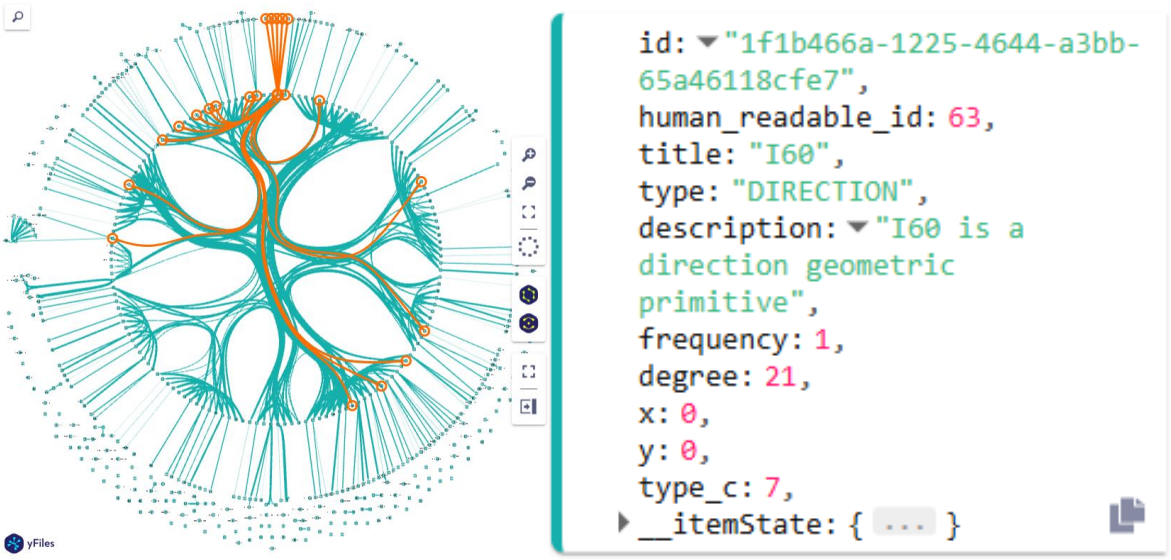


Рис. 3. Граф сборки.

Значения метрик конвейера GraphRAG на наборе из 198 вопросов: Answer Relevancy – 0.483, Context Recall – 0.727, Context Relevance – 0.975, Answer Accuracy – 0.486.

Необходимо сделать уточнение в части алгоритма формирования контекста при расчете метрик Context Recall и Context Relevance. Эти метрики используют в качестве входных данных извлеченные контексты и нацелены на

оценку фазы извлечения конвейера RAG. Однако при формировании ответа GraphRAG фаза извлечения радикально отличается от конвейера RAG и предусматривает формирование выборки релевантных сущностей, связей и обобщений по кластерам [19]. Поэтому для расчета показателей метрик конвейера GraphRAG в контекст были включены значения сущностей и связей, что привело к кратному отличию значений метрики Context Recall и значительному различию в метрике Context Relevance, обусловленному разницей в алгоритме формирования контекста.

3. РЕЗУЛЬТАТЫ

Было выполнено сравнение значений метрик конвейеров RAG и GraphRAG на полученном датасете. Проведенное сравнение показателей метрик конвейеров RAG и GraphRAG продемонстрировало положительный вклад извлечения структуры данных в GraphRAG-подходе. Приросты значений метрик составили (табл. 2): Answer Relevancy + 0.124, Context Recall + 0.611, Context Relevance + 0.424, Answer Accuracy + 0.119. Наиболее показательными (используют только вопросы и тексты ответов) являются сравнения метрик Answer Accuracy + 32.3% и Answer Relevancy + 34.7% в пользу GraphRAG.

Табл. 2. Сравнение значений метрик конвейеров RAG и GraphRAG.

Метрика	RAG	GraphRAG	Разница	Лучший подход
Answer Accuracy	0.367	0.486	0.119	GraphRAG
Answer Relevancy	0.359	0.483	0.124	GraphRAG
Context Recall	0.116	0.727	0.611	GraphRAG
Context Relevance	0.551	0.975	0.424	GraphRAG

Различия принципов формирования контекста в RAG- и GraphRAG-подходах приводят к объективным различиям в объеме и содержании контекста. В контекст запросов в RAG попадает ограниченное число чанков на основании семантического сходства, что ограничивает полноту извлечения связанных аксиом. В контекст запросов GraphRAG включаются таблицы релевантных сущностей и связей, обобщения, а также чанки документов. Такой подход к наполнению контекста может быть избыточным при формировании

ответов в доменах с ограниченным набором связей (например, в художественной литературе), однако в домене машиностроительной PLM на данных со связной логической структурой обеспечивает прирост качества.

В результате проведенных исследований:

– предложен новый источник оценочных данных в домене машиностроительных систем полного жизненного цикла – файлы изделий и сборок, соответствующих стандарту *STandard for Exchange of Product model data*;

– предложена новая методика формирования текстовых оценочных данных с сохранением логической структуры посредством использования онтологических данных об изделиях в машиностроительных системах полного жизненного цикла и последующей вербализацией с применением однозначного преобразования на управляемый английский – Attempto Controlled English;

– реализован конвейер для использования сгенерированных оценочных данных при сравнении систем обогащения контекста большой языковой модели с применением и без применения графов знаний;

– получены результаты оценки, подтверждающие положительный вклад систем с извлечением структурированных данных в качество генерируемых ответов в доменной области машиностроительных систем полного жизненного цикла.

ЗАКЛЮЧЕНИЕ

Разработана методика генерации синтетического набора данных на основе формальной онтологии, полученной из STEP- файлов – стандарта обмена данными изделий систем автоматизированного проектирования и PLM- систем. Для обеспечения отображения онтологии в текстовую модальность с гарантированным сохранением логической структуры использовано формальное преобразование на язык ACE, что обеспечивает объективную основу для оценки качества извлечения знаний.

Ключевым результатом является формирование методики построения набора данных в доменной области с значительным дефицитом оценочных данных. Такая методика реализует детерминированное формирование набора

данных (свободное от влияния вероятностных факторов при генерации с помощью БЯМ) в автоматическом режиме с возможностью масштабирования.

Практическая значимость работы заключается в реализации программного конвейера для сравнения и оценки извлечения графов знаний с помощью GraphRAG в доменной области машиностроительной PLM. Этот конвейер позволяет получать измеримую оценку для повышения качества обработки структурированной информации, что имеет критическое значение для сохранения логических связей, характерных для данных PLM-домена.

Получены результаты оценки, подтверждающие положительный вклад в качество генерируемых ответов систем, использующих структуру данных (GraphRAG), по сравнению с конвейерами, не использующими извлечение структуры данных (традиционные RAG), в доменной области машиностроительных PLM.

СПИСОК ЛИТЕРАТУРЫ

1. *Edge D. et al.* From local to global: A graph rag approach to query-focused summarization // arXiv: 2404.16130. https://doi.org/10.48550/arXiv.2404.16130

2. *Xiang Z. et al.* When to use Graphs in RAG: A Comprehensive Analysis for Graph Retrieval-Augmented Generation // arXiv: 2506.05690. https://doi.org/10.48550/arXiv.2506.05690

3. *Han H. et al.* Rag vs. graphrag: A systematic Evaluation and Key Insights // arXiv: 2502.11371. 2025. https://doi.org/10.48550/arXiv.2502.11371

4. *Han H. et al.* Retrieval-Augmented Generation with Graphs (GraphRAG) // arXiv: 2501.00309. 2024. https://doi.org/10.48550/arXiv.2501.00309

5. *Gajderowicz B., Bhardwaj A., Fox M.* RAG and Ontologies for Information Retrieval: A Literature Review. 2025. URL: https://eil.mie.utoronto.ca/wp-content/uploads/2025/09/ONTOLLM_2025_Aug20v11.pdf (дата обращения: 12.02.2026)

6. *Huang Y., Fung T.Y., DeLaurentis D.A.* Addressing Complexity in System of Systems with GraphRAG: An AI-Driven Framework for Dynamic Data Integration // Systems Engineering. 2025. e70012. https://doi.org/10.1002/sys.70012

7. *da Cruz T., Tavares B., Belo F.* Ontology Learning and Knowledge Graph Construction: A Comparison of Approaches and Their Impact on RAG Performance // arXiv: 2511.05991. 2025. https://doi.org/10.48550/arXiv.2511.05991

8. ГОСТ Р 56862–2016. Национальный стандарт Российской Федерации «Система управления жизненным циклом. Разработка концепции изделия и технологий. Термины и определения».
URL: https://protect.gost.ru/document.aspx?control=7&id=202875 (дата обращения: 12.02.2026)

9. ГОСТ Р ИСО 10303-1 – 2022. Национальный стандарт Российской Федерации «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы (ISO 10303-1:2021, IDT)»
URL: https://meganorm.ru/Data/792/79232.pdf (дата обращения: 12.02.2026)

10. *Kwon S., Monnier L.V., Barbau R., Bernstein W.Z.* A New Implementation of OntoSTEP: Flexible Generation of Ontology and Knowledge Graphs of EXPRESS-Driven Data // ASME Journal of Computing and Information Science in Engineering. 2022. Vol. 22 (2). 024502. https://doi.org/10.1115/1.4053079

11. Stepcode.
URL: https://github.com/stepcode/stepcode/tree/develop/ data/ ap214e3 (дата обращения: 12.02.2026)

12. *Zhang Q. et al.* A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models // arXiv: 2501.13958. 2025.
https://doi.org/10.48550/arXiv.2501.13958

13. *Fuchs N.E., Kaljurand K., Kuhn T.* Attempto controlled english for knowledge representation //Reasoning Web: 4th International Summer School 2008, Venice, Italy, September 7–11, 2008, Tutorial Lectures. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. P. 104–124. https://doi.org/10.1007/978-3-540-85658-0

14. *Zaitoun A., Sagi T., Peleg M.* Generating ontology-learning training-data through verbalization //Proc. of the AAAI Symposium Series. 2024. Vol. 4. No. 1. P. 233–241. https://doi.org/10.1609/aaaiss.v4i1.31797

15. *Fuchs N.E. et al.* Attempto controlled english: A knowledge representation language readable by humans and machines // Reasoning Web: First International Summer School 2005. Springer Berlin Heidelberg, 2005. P. 213–250. https://doi.org/10.1007/11526988_6

16. *Dong S. et al.* Knowledge-Graph Based RAG System Evaluation Framework // arXiv: 2510.02549. 2025. https://doi.org/10.48550/arXiv.2510.02549

17. *Chen E. et al.* Comparing RAG and GraphRAG for Page-Level Retrieval Question Answering on Math Textbook // arXiv: 2509.16780. 2025. https://doi.org/10.48550/arXiv.2501.13958

18. *Xiao Y. et al.* GraphRAG-Bench: Challenging Domain-Specific Reasoning for Evaluating Graph Retrieval-Augmented Generation // arXiv: 2506.02404. 2025. https://doi.org/10.48550/arXiv.2506.02404

19. *Es S. et al.* Ragas: Automated evaluation of retrieval augmented generation // Proc. of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024. P. 150–158. https://doi.org/10.48550/arXiv.2309.15217

ONTOLOGICAL APPROACH TO KNOWLEDGE GRAPH ASSESSMENT IN THE DOMAIN OF MECHANICAL PRODUCT LIFECYCLE MANAGEMENT SYSTEMS

V. V. Gladyshev[0009-0003-8900-3469]

Moscow Institute of Physics and Technology, The Pusk Center, Dolgoprudny, Russia

gladyshev.vv@phystech.edu

Abstract

This paper examines the application of an ontological approach to constructing a dataset for evaluating and comparing context enrichment systems for large language models using knowledge graphs in the domain of mechanical product lifecycle management systems. In this domain, obtaining the required amount of textual data with a formal logical structure to form an evaluation set without using generated synthetic data is challenging. To avoid introducing distortions and hallucinations when generating evaluation data, a novel solution to the data deficiency is proposed. This solution involves extracting ontology directly from product and assembly files compliant with the STandard for Exchange of Product Model Data. This potentially enables the use of all product data as a source for scaling evaluation data. The goal of this paper is to create a dataset of structured textual data in the domain of mechanical product lifecycle management systems, develop an evaluation methodology, and implement context enrichment pipelines for large language models with and without knowledge graphs to analyze the contribution of data-structure-extracting systems to the quality of generated responses. In this paper: a new source of evaluation data is proposed, a new methodology for generating text evaluation data while preserving the logical structure is developed, a pipeline for using the generated evaluation data is implemented, and evaluation results are obtained that confirm the positive contribution of systems with the extraction of structured data to the quality of generated responses in the domain of mechanical product lifecycle management systems.

Keywords**:** *ontologies, dataset, product lifecycle management – PLM, computer-aided design – CAD, large language model – LLM, retrieval-augmented generation – RAG, GraphRAG, STandard for Exchange of Product model data – STEP.*

REFERENCES

1. *Edge D. et al.* From local to global: A graph rag approach to query-focused summarization // arXiv: 2404.16130. https://doi.org/10.48550/arXiv.2404.16130

2. *Xiang Z. et al.* When to use Graphs in RAG: A Comprehensive Analysis for Graph Retrieval-Augmented Generation // arXiv: 2506.05690. https://doi.org/10.48550/arXiv.2506.05690

3. *Han H. et al.* Rag vs. graphrag: A systematic Evaluation and Key Insights // arXiv: 2502.11371. 2025. https://doi.org/10.48550/arXiv.2502.11371

4. *Han H. et al.* Retrieval-Augmented Generation with Graphs (GraphRAG) // arXiv: 2501.00309. 2024. https://doi.org/10.48550/arXiv.2501.00309

5. *Gajderowicz B., Bhardwaj A., Fox M.* RAG and Ontologies for Information Retrieval: A Literature Review. 2025. URL: https://eil.mie.utoronto.ca/wp-content/uploads/2025/09/ONTOLLM_2025_Aug20v11.pdf (last access: 12.02.2026)

6. *Huang Y., Fung T.Y., DeLaurentis D.A.* Addressing Complexity in System of Systems With GraphRAG: An AI-Driven Framework for Dynamic Data Integration // Systems Engineering. 2025. e70012. https://doi.org/10.1002/sys.70012

7. *da Cruz T., Tavares B., Belo F.* Ontology Learning and Knowledge Graph Construction: A Comparison of Approaches and Their Impact on RAG Performance // arXiv: 2511.05991. 2025. https://doi.org/10.48550/arXiv.2511.05991

8. GOST R 56862-2016. Nacional'nyj standart Rossijskoj Federacii “Sistema upravlenia ziznennym ciklom. Razrabotka koncepcii izdelia i tehnologij. Terminy i opredelenia”. URL: https://protect.gost.ru/document.aspx?control=7&id=202875 (last access: 12.02.2026)

9. GOST R ISO 10303-1-2022. Nacional'nyj standart Rossijskoj Federacii “Sistemy avtomatizacii proizvodstva i ih integracia. Predstavlenie dannyh ob izdelii i obmen etimi dannymi. Cast' 1. Obsie predstavlenia i osnovopolagausie principy (ISO 10303-1:2021, IDT)” URL: https://meganorm.ru/Data/792/79232.pdf (last access: 12.02.2026)

10. *Kwon S., Monnier L.V., Barbau R., Bernstein W.Z.* A New Implementation of OntoSTEP: Flexible Generation of Ontology and Knowledge Graphs of EXPRESS-Driven Data // ASME Journal of Computing and Information Science in Engineering. 2022. Vol. 22 (2). 024502. https://doi.org/10.1115/1.4053079

11. Stepcode. URL: https://github.com/stepcode/stepcode/tree/develop/data/ ap214e3 (last access: 12.02.2026)

12. *Zhang Q. et al.* A Survey of Graph Retrieval-Augmented Generation for Customized Large Language Models // arXiv: 2501.13958. 2025. https://doi.org/10.48550/arXiv.2501.13958

13. *Fuchs N.E., Kaljurand K., Kuhn T.* Attempto controlled english for knowledge representation //Reasoning Web: 4th International Summer School 2008, Venice, Italy, September 7–11, 2008, Tutorial Lectures. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. P. 104–124. https://doi.org/10.1007/978-3-540-85658-0

14. *Zaitoun A., Sagi T., Peleg M.* Generating ontology-learning training-data through verbalization //Proc. of the AAAI Symposium Series. 2024. Vol. 4. No.1. P. 233–241. https://doi.org/10.1609/aaaiss.v4i1.31797

15. *Fuchs N.E. et al.* Attempto controlled english: A knowledge representation language readable by humans and machines // Reasoning Web: First International Summer School 2005. Springer Berlin Heidelberg, 2005. P. 213–250. https://doi.org/10.1007/11526988_6

16. *Dong S. et al.* Knowledge-Graph Based RAG System Evaluation Framework // arXiv: 2510.02549. 2025. https://doi.org/10.48550/arXiv.2510.02549

17. *Chen E. et al.* Comparing RAG and GraphRAG for Page-Level Retrieval Question Answering on Math Textbook // arXiv: 2509.16780. 2025. https://doi.org/10.48550/arXiv.2501.13958

18. *Xiao Y. et al.* GraphRAG-Bench: Challenging Domain-Specific Reasoning for Evaluating Graph Retrieval-Augmented Generation // arXiv: 2506.02404. 2025. https://doi.org/10.48550/arXiv.2506.02404

19. *Es S. et al.* Ragas: Automated evaluation of retrieval augmented generation // Proc. of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024. P. 150–158. https://doi.org/10.48550/arXiv.2309.15217

СВЕДЕНИЯ ОБ АВТОРЕ

ГЛАДЫШЕВ Виталий Владимирович – магистрант МФТИ, Центр «Пуск». Область научных интересов: Средства обработки естественного языка и автоматизации структурирования данных, онтологии, применение больших языковых моделей и технологии их адаптации, RAG, GraphRAG, агентные системы.

Vitaly Vladimirovich GLADYSHEV – master's student at MIPT, the Pusk Center. Research interests: Natural language processing and automation of data structuring, ontologies, application of large language models and technologies for their adaptation, RAG, GraphRAG, agent systems.

email: gladyshev.vv@phystech.edu

ORCID: 0009-0003-8900-3469

Материал поступил в редакцию 15 апреля 2026 года

УДК 621.372

ИНТЕЛЛЕКТУАЛЬНЫЙ АССИСТЕНТ ДЛЯ ПРОЕКТИРОВАНИЯ ЭКРАНОВ РАДИАЦИОННОЙ ЗАЩИТЫ

Л. А. Зинченко[1] [0000-0002-2298-8721], **А. М. Чернецов**[2] [0000-0001-7655-2395], **В. В. Казаков**[3] [0000-0003-1571-0104], **Е. С. Поляков**[4] [0009-0007-8136-383X], **Е. Н. Комкова**[5] [0009-0003-9776-2094], **В. М. Киселева**[6] [0009-0002-6591-6186]

[1, 3–6]Московский государственный технический университет им. Н.Э. Баумана, *г. Москва, Россия*

[2]*Национальный исследовательский университет «МЭИ», г. Москва, Россия*

[3]ООО БКС «Финтех», *г. Москва, Россия*

[1]lyudmillaa@mail.ru, [2]chernetsovam@mpei.ru, [3]kazakov.vadim.2012@yandex.ru, [4]polyakoves@student.bmstu.ru, [5]komkovaen@student.bmstu.ru, [6]kiselevavm@student.bmstu.ru

Аннотация

Рассмотрена актуальная задача разработки интеллектуального агента для моделирования характеристик экранов защиты электронной аппаратуры, который позволит упростить анализ различных проектных решений и обеспечить поддержку принятия решения инженером-проектировщиком. Разработан интеллектуальный агент, позволяющий автоматизировать процесс подготовки описания альтернативного проектного решения для последующего моделирования с использованием программного пакета Geant4. Интеграция программного модуля в вычислительные платформы даст возможность усовершенствовать работу инженера-проектировщика за счет сокращения рутинных ручных операций, минимизировать человеческие ошибки и гарантировать воспроизводимость результатов.

Ключевые слова*: интеллектуальный ассистент, агент, генеративные технологии, автоматизация, радиация, тяжелые заряженные частицы, моделирование, Geant4.*

ВВЕДЕНИЕ

При проектировании изделий, предназначенных для использования в космическом пространстве, необходимо предусмотреть защиту аппаратуры от воздействия космической радиации [1, 2]. Одним из возможных способов является применение экранов радиационной защиты. При их проектировании необходимо моделирование прохождения частиц через материал экрана. Для решения этой задачи используются различные программные пакеты [3–5]. В настоящей работе рассмотрен подход к созданию интеллектуального ассистента, применение которого позволит автоматизировать ручной этап подготовки задания для программного пакета Geant4 [6–8].

МЕТОДЫ ОЦЕНКИ СТЕПЕНИ ЗАЩИТЫ АППАРАТУРЫ С ПОМОЩЬЮ ЭКРАНОВ

Для моделирования прохождения частиц через материалы используются различные программные пакеты. Одним из наиболее популярных пакетов является SRIM/TRIM [4, 5]. Эта программа помогает рассчитать траекторию и потери энергии ионов в веществе, а также результирующее воздействие радиации на материал. Инструментарий SRIM/TRIM широко используется в различных областях исследований, таких как изучение радиационной стойкости вещества, медицинская физика и решение прикладных инженерных задач. Он также применяется при проектировании атомных электростанций и расчете аппаратуры для освоения космоса.

В [4–6] SRIM/TRIM использован для моделирования экранов радиационной защиты электронного оборудования. В [5] описан программный модуль для автоматизированного расчета эффективности многослойных экранов для защиты бортового электронного оборудования от воздействия тяжелых заряженных частиц на основе SRIM. Реализована клиент-серверная архитектура с веб-интерфейсом, позволяющая осуществлять распределенные вычисления. Этот модуль автоматизирует генерацию входных файлов для SRIM, выполняет вычисления и обрабатывает результаты. Такой маршрут значительно сокращает трудозатраты на проектирование. Однако инструментарий SRIM имеет ряд недостатков, включая сложность конфигурации для выполнения вычислений, ограниченную визуализацию результатов, а также отсутствие явных данных о средней

глубине проникновения частиц. Кроме того, в программном решении, описанном в [4, 5], нет модуля интеллектуального ассистента, который позволял бы упростить взаимодействие инженера-проектировщика с пакетом SRIM/TRIM.

Наше программное обеспечение использует программный пакет Geant4 [7–10], который поддерживает большее количество библиотек и обеспечивает более корректное моделирование прохождения частиц через вещество. Geant4 позволяет пользователям настраивать компоненты моделирования в соответствии с конкретными задачами.

ПРЕДЛАГАЕМЫЕ ПОДХОДЫ

Разработанное программное обеспечение предоставляет возможность выбора двух маршрутов проектирования: ручное, без использования интеллектуального ассистента и автоматическую подготовку описания альтернативного проектного решения с использованием интеллектуального ассистента. На рис. 1 представлена схема взаимодействия инженера-проектировщика и интеллектуального ассистента.

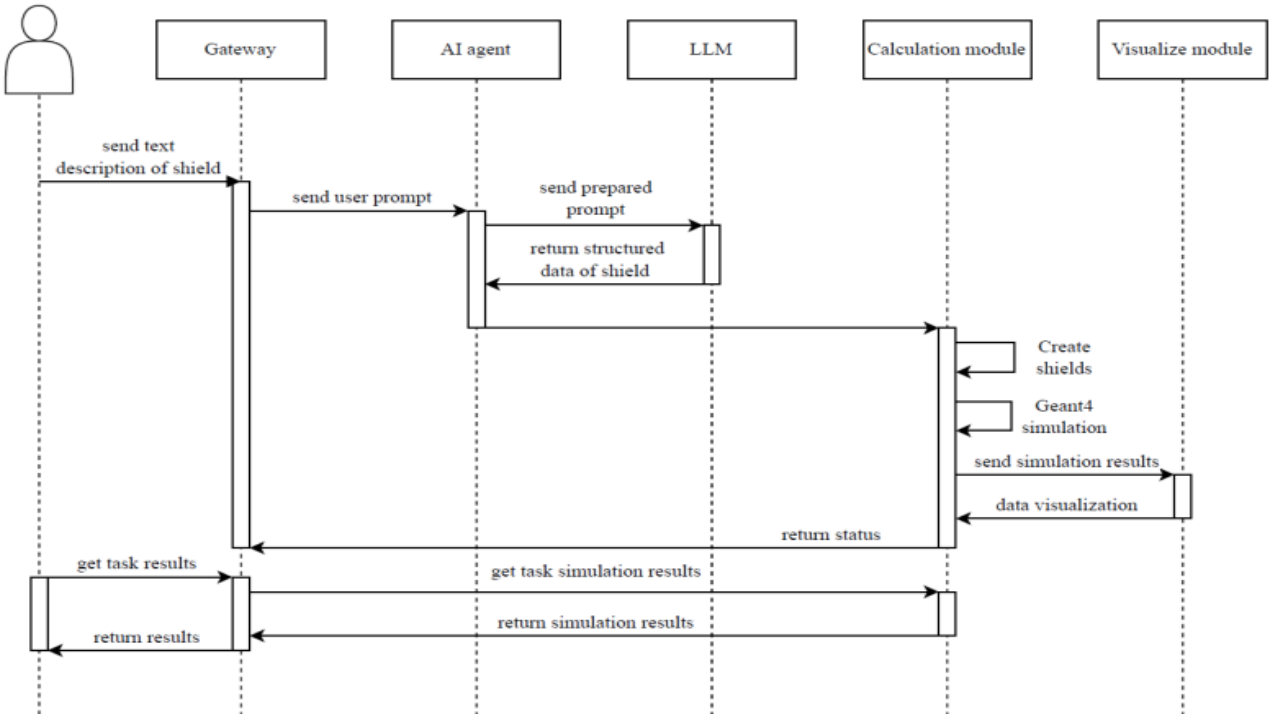


Рис. 1. Диаграмма взаимодействия инженера-проектировщика и интеллектуального ассистента.

В результате этого взаимодействия генерируются промпт и описание альтернативного проектного решения для последующего моделирования с использованием пакета Geant4.

Была использована языковая модель GigaChat [11]. Пользователь на естественном языке описывает задачу, используя инженерную терминологию. Затем агент преобразует задачи проектирования в файл формата JSON на основе запроса пользователя, что значительно сокращает трудозатраты по сравнению с ручным вводом данных.

СРАВНЕНИЕ РАЗРАБОТАННЫХ ПОДХОДОВ

Для проверки работоспособности всех модулей разработанного программного обеспечения были выполнены различные тесты как для ручного маршрута, так и для маршрута с использованием интеллектуального ассистента.

В тесте для ручного маршрута экран состоял из слоев бериллия и алюминия. С ним взаимодействуют несколько типов частиц с разной энергией: ^{3}He – ядра изотопа гелия–3 с энергией 60 МэВ, протоны с энергией 30 МэВ и электроны с энергией 20 МэВ. В табл. 1 представлены результаты моделирования. Экран задерживает частицы изотопа гелия, но электроны проходят через экран. Наблюдается также вторичное излучение при прохождении электронов. Частицы протонов частично задерживаются.

Табл. 1. Результаты моделирования для ручного маршрута.

Тип частиц	Энергия, МэВ	Материал (толщина, мм)	Общее количество вторичных частиц	Количество остановившихся первичных частиц	Количество остановившихся вторичных частиц
He^3	60	Бериллий (2.7)	0	10	1
		Алюминий (2.0)		0	0
Электрон	20	Бериллий (2.7)	100	0	5
		Алюминий (2.0)		0	3

Протон	30	Бериллий (2.7)	425	0	1
		Алюминий (2.0)		5	0

При использовании интеллектуального ассистента пользователь генерирует описание конфигурации экрана в виде соответствующего запроса. На основе этого запроса ассистент генерирует описание модели. В тесте экран состоял из слоев бериллия и сплава ВТ5Л. В этом тесте было исследовано прохождение частицы изотопа гелия с энергией 60 МэВ.

Во втором тесте пользователь формирует описание конфигурации экрана в виде соответствующего запроса на естественном языке, представленного на рис. 2.

```
Помоги подготовить информацию для экрана. Помоги
составить экран из слоев Be и ВТ5Л.
Первый слой толщиной 1000 мкм, второй 2000 мкм.
Найди необходимые данные. Укажи правильные символы
элементов, атомный номер, стандартный атомный вес
элементов, плотность.
```

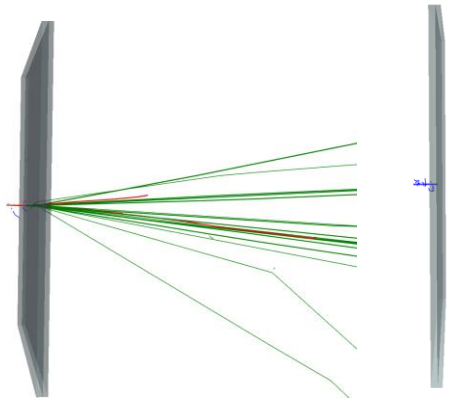
Рис. 2. Пример запроса пользователя описания экрана.

В запросе пользователь перечисляет слои конфигурируемого экрана, а также соответствующие названия материалов и толщины. На основе этого текстового описания агент искусственного интеллекта выполнил автономную генерацию входных данных для моделирования: самостоятельно определил химический состав каждого слоя, подобрал необходимые физические параметры материалов и сформировал структурированную конфигурацию в требуемом формате. В данном тесте сгенерированный агентом экран облучается частицами изотопа гелия-3 с энергией 60 МэВ. Результаты моделирования показаны в табл. 2.

Табл. 2. Результаты моделирования с использованием интеллектуального ассистента.

Тип частиц	Энергия, МэВ	Материал (толщина, мм)	Общее количество вторичных частиц	Количество остановившихся первичных частиц	Количество остановившихся вторичных частиц
He^3	60	Бериллий (1.0)	3	0	3
		Сплав ВТ5Л (2.0)		20	0

На основе полученных данных можно заметить, что частицы останавливаются в слое ВТ5Л, в то время как слой бериллия останавливает вторично генерируемые частицы. Визуализация тестов представлена на рис. 3 (а, б).



а) б)

Рис. 3. Визуализация результатов маршрута с использованием: а) ручного маршрута; б) интеллектуального ассистента.

Возможность выбрать предпочтительный вариант (ручное управление или управление с помощью искусственного интеллекта) является существенным преимуществом нашего программного обеспечения благодаря гибкости и адаптивности для пользователей с различными предпочтениями.

ЗАКЛЮЧЕНИЕ

Проведен анализ некоторых из возможных подходов для автоматизации проектирования экранов радиационной защиты. Показано, что для использования сложного специализированного программного обеспечения с целью упрощения его применения возможно использование интеллектуального агента на базе большой языковой модели. Полученные результаты определяют дальнейшие перспективные направления исследований в области применения систем искусственного интеллекта и агентов на их основе в задачах разработки систем автоматизированного проектирования отечественных роботизированных и высокопроизводительных вычислительных систем, в том числе предназначенных для исследования космоса.

СПИСОК ЛИТЕРАТУРЫ

1. *Shakhnov V., Zinchenko L., Kosolapov I., Filippov I.* Modeling and Optimization of Radiation Tolerant Microsystems // EMS'14 Proceedings. 2014. P. 484–489.

2. *Glushko A.A., Morozov S.A., Chistyakov M.G.* Study of the Sensitive Region of a MOS Transistor to the Effects of Secondary Particles Arising from Ionizing Radiation // Microelectronics. 2023. Vol. 52. No. 4. P. 282–289.

3. *Terekhov V.V., Glushko A.A., Makarchuk V.V. et al.* Compact modeling and digital twins of capacitive fractal microsystems: characteristics variations caused by heavy charged particle // REEPE. 2023. P. 1–5. https://doi.org/10.1109/REEPE57272.2023.10086770

4. *Glushko A.A., Zinchenko L.A., Shakhnov V.A.* Simulation of the impact of heavy charged particles on the characteristics of field-effect silicon-on-insulator transistors // Journal of Communications Technology and Electronics. 2015. Vol. 60. P. 1134–1140. https://doi.org/10.1134/S1064226915070074

5. *Zinchenko L., Kazakov V., Mironov A. et al.* Software module for automated calculation of parameters of on-board electronic equipment protection screens from radiation exposure // Journal of Software & Systems. 2020. P. 236–242

6. *Shakhnov V.A., Zinchenko L.A., Rezchikova E.V. et al.* Visual Analytics and Its Applications in Electronic Engineering Education: BMSTU Case Study // Proceedings

of VI International Conference on Information Technologies in Engineering Education, Inforino 2022. https://doi.org/10.1109/Inforino53888.2022.9782907

7. *Allison J., Amako K., Apostolakis J. et al.* Recent developments in Geant4. Nuclear Instruments and Methods in Physics Research // Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2016. Vol. 835. P. 186–225.

8. *Allison J., Amako K., Apostolakis J., Araujo H., Arce Dubois P.* Geant4 developments and applications // IEEE Transactions on Nuclear Science. 2006. Vol. 53. P. 270–278. https://doi.org/10.1109/TNS.2006.869826

9. *Agostinelli S., Allison J., Amako K. et al.* Geant4 – a simulation toolkit // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2003. V. 506. P. 250–303.

10. *Зинченко Л.А., Казаков В.В., Моисеев Р.Р. и др.* Программный модуль для автоматизированного проектирования многослойных экранов защиты электронной аппаратуры от воздействия тяжелых заряженных частиц с использованием Geant4 // Известия ЮФУ. Технические науки. 2024.

11. GigaChat: API GigaChat.
URL: https://developers.sber.ru/docs/ru/gigachat/api/overview (дата обращения: 09.12.2025).

AI COPILOT FOR DESIGNING RADIATION PROTECTION SHIELD

L. A. Zinchenko[1] [0000-0002-2298-8721], A. M. Chernetsov[2] [0000-0001-7655-2395],
V. V. Kazakov[3] [0000-0003-1571-0104], E. S. Polyakov[4] [0009-0007-8136-383X],
E. N. Komkova[5] [0009-0003-9776-2094], V. M. Kiseleva[6] [0009-0002-6591-6186]

[1, 3–6]Bauman Moscow State Technical University, Moscow, Russia
[2]Moscow Power Engineering Institute, Moscow, Russia
[3]BCS Fintech LLC, Moscow, Russia

[1]lyudmilaaa@mail.ru, [2]chernetsovam@mpei.ru, [3]kazakov.vadim.2012@yandex.ru,
[4]polyakoves@student.bmstu.ru, [5]komkovaen@student.bmstu.ru,
[6]kiselevavm@student.bmstu.ru

Abstract

This paper examines the current challenge of developing an intelligent agent for modeling the characteristics of electronic equipment protection shields.

The aim is developing a methodology and software implementation for an intelligent agent that will simplify the analysis of various design solutions and provide decision support for design engineers. An intelligent agent has been developed that automates the process of preparing a description of an alternative design solution for subsequent modeling using the Geant4 software package. Integrating the software module into computing platforms will improve the work of design engineers by reducing routine manual operations, minimizing human error, and ensuring reproducible results.

Keywords*: AI Copilot, agentic AI, generative technologies, automation, radiation, heavy charged particles, modelling, Geant4.*

REFERENCES

1. *Shakhnov V., Zinchenko L., Kosolapov I., Filippov I.* Modeling and Optimization of Radiation Tolerant Microsystems // EMS'14 Proceedings. 2014. P. 484–489.

2. *Glushko A.A., Morozov S.A., Chistyakov M.G.* Study of the Sensitive Region of a MOS Transistor to the Effects of Secondary Particles Arising from Ionizing Radiation // Microelectronics. 2023. Vol. 52. No. 4. P. 282–289.

3. *Terekhov V.V., Glushko A.A., Makarchuk V.V. et al.* Compact modeling and digital twins of capacitive fractal microsystems: characteristics variations caused by heavy charged particle // REEPE 2023. P. 1–5.
https://doi.org/10.1109/REEPE57272.2023.10086770

4. *Glushko A.A., Zinchenko L.A., Shakhnov V.A.* Simulation of the impact of heavy charged particles on the characteristics of field-effect silicon-on-insulator transistors // Journal of Communications Technology and Electronics. 2015. Vol. 60. P. 1134–1140. https://doi.org/10.1134/S1064226915070074

5. *Zinchenko L., Kazakov V., Mironov A. et al.* Software module for automated calculation of parameters of on-board electronic equipment protection screens from radiation exposure // Journal of Software & Systems. 2020. P. 236–242.

6. *Shakhnov V.A., Zinchenko L.A., Rezchikova E.V. et al.* Visual Analytics and Its Applications in Electronic Engineering Education: BMSTU Case Study // Proceedings of VI International Conference on Information Technologies in Engineering Education, Inforino 2022. https://doi.org/10.1109/Inforino53888.2022.9782907

7. *Allison J., Amako K., Apostolakis J. et al.* Recent developments in Geant4. Nuclear Instruments and Methods in Physics Research // Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2016. Vol. 835. P. 186–225.

8. *Allison J., Amako K., Apostolakis J., Araujo H., Arce Dubois P.* Geant4 developments and applications // IEEE Transactions on Nuclear Science. 2006. Vol. 53. P. 270–278. https://doi.org/10.1109/TNS.2006.869826

9. *Agostinelli S., Allison J., Amako K.* et al. Geant4 – a simulation toolkit // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2003. Vol. 506. P. 250–303.

10. *Zinchenko L.A., Kazakov V.V., Moiseev R.R. et al.* Software module for computer-aided design of multilayer screens protecting electronic equipment from the effects of heavy charged particles using Geant4 // Izvestiya SFEDU. Technical Sciences. 2024. P. 100–109.

11. GigaChat: API GigaChat.
URL: https://developers.sber.ru/docs/ru/gigachat/api/overview. (date of access: 09.12.2025).

СВЕДЕНИЯ ОБ АВТОРАХ

ЗИНЧЕНКО Людмила Анатольевна – доктор техн. наук, профессор, профессор МГТУ им. Н.Э. Баумана

Lyudmila Anatolyevna ZINCHENKO – Doctor of Technical Sciences, Professor, professor at the Department IU BMSTU

email: lyudmillaa@mail.ru

ORCID: 0000-0002-2298-8721

ЧЕРНЕЦОВ Андрей Михайлович – кандидат технических наук, доцент; доцент кафедры Прикладной математики и искусственного интеллекта Национального исследовательского университета «МЭИ».

Andrey Mikhailovich CHERNETSOV – Candidate of Technical Sciences, Associate Professor; associate professor at the Department of Applied Mathematics and Artificial Intelligence, National Research University "MPEI".

email: chernetsovam@mpei.ru

ORCID: 0000-0001-7655-2395

КАЗАКОВ Вадим Вячеславович – кандидат технических наук, доцент МГТУ им. Н.Э. Баумана; руководитель направления по искусственному интеллекту ООО БКС «ФИНТЕХ».

Vadim Vyacheslavovich KAZAKOV – Candidate of Technical Sciences, Ass. Professor, BMSTU; pead of the Artificial Intelligence department at BCS FINTECH LLC.

email: kazakov.vadim.2012@yandex.ru

ORCID: 0000-0003-1571-0104

Поляков Евгений Сергеевич – студент, МГТУ им. Н.Э. Баумана.

Evgeniy Sergeevich Polyakov –student, BMSTU

email: polyakoves@student.bmstu.ru

ORCID: 0009-0007-8136-383X

КИСЕЛЕВА Валентина Михайловна – студент, МГТУ им. Н.Э. Баумана.

Valentina Mikhailovna KISELEVA –student, BMSTU
email: kiselevavm@student.bmstu.ru
ORCID: 0009-0002-6591-6186

КОМКОВА Елена Николаевна – студент, МГТУ им. Н.Э. Баумана.

Elena Nikolaevna KOMKOVA – student, BMSTU
email: komkovaen@student.bmstu.ru
ORCID: 0009-0003-9776-2094

Материал поступил в редакцию 14 апреля 2026 года

УДК 004.4

РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ПОИСКА СЕМАНТИЧЕСКИ БЛИЗКИХ ФРАГМЕНТОВ ПРОГРАММНОГО КОДА

В. И. Зорин[1] [0009-0004-0271-1882], Е. К. Липачев[2] [0000-0001-7789-2332]

[1]*Казанский национальный исследовательский технический университет имени А. Н. Туполева — КАИ, г. Казань,* Россия
[2]*Казанский (Приволжский) федеральный университет, г. Казань,* Россия
[2]*Университет Иннополис, г. Иннополис, Россия*

[1]addefan@mail.ru, [2]elipachev@gmail.com

Аннотация

Рекомендательные системы в научном информационном пространстве являются инструментом поиска и навигации при работе с научными документами. Программный код в настоящее время рассматривается как объект научного знания, и, как следствие, важной задачей является создание систем поддержки жизненного цикла программ, в частности поиска близких программных решений, обнаружения заимствований программного кода, анализа и оценки качества кода. В работе предложена рекомендательная система, формирующая для пользователя персонализированный список фрагментов кода, функционально эквивалентных входному коду-запросу, представленному на одном из языков программирования из установленного набора. Базовый алгоритм системы основан на представлении программного кода в виде абстрактного синтаксического дерева с последующим построением векторного пространства программных кодов. Семантическое сходство программных кодов определяется по расстоянию между векторами кодов в многомерном пространстве. Персонализация выдачи достигается за счет модуля фильтрации, который ранжирует найденные фрагменты с учетом профиля пользователя. Рассматриваемыми факторами являются языковые предпочтения пользователя и его области научных интересов, извлекаемые посредством интеграции с ORCID. Для обеспечения работы системы на основе корпуса CodeNet создан специализированный набор фрагментов про-

граммного кода. Решена также задача автоматического определения языка программирования по фрагменту представленного кода на одном из языков, входящих в текущий рейтинговый список языков программирования.

***Ключевые слова**: абстрактное синтаксическое дерево, векторизация кода, контентная фильтрация, кросс-языковой поиск, межъязыковой программный клон, рекомендательная система, сходство программного кода.*

ВВЕДЕНИЕ

В условиях стремительного усложнения программных систем и расширения цифровых научных архивов разработка методов кросс-языкового анализа исходных кодов программ становится приоритетным направлением исследований. Под семантической близостью программных фрагментов в контексте настоящей работы понимается сходство их функционального поведения, которое аппроксимируется совокупностью структурных и синтаксических паттернов, отражающих логику обработки данных. Под межъязыковым (кросс-языковым) клоном будем понимать фрагмент кода, который демонстрирует синтаксическое (текстовое и структурное) или семантическое (функциональное) сходство с фрагментом кода, написанным на другом языке программирования.

Решение задачи поиска близкого программного кода имеет и существенную практическую ценность в оптимизации процессов обучения и промышленной разработке программного обеспечения. Возможность сопоставления алгоритмически эквивалентных решений на различных языках позволяет значительно ускорить освоение новых технологических идиом и упростить поддержку гетерогенных проектов, доля которых, по статистике, в современной индустрии превышает 80% (см., например, [1]). Кроме того, такие инструменты критически важны при глубоком рефакторинге и миграции систем на новые наборы технологий.

Несмотря на интенсивное развитие области кросс-языкового поиска клонов, большинство существующих инструментов, созданных в рамках соответствующих исследований (см., например, [2–6]), характеризуется жесткой привязкой к ограниченному набору синтаксисов или недостаточной глубиной анализа семантических связей.

Задачу поиска близкого по содержанию кода исследуют преимущественно в контексте поиска клонов кода (выявление дубликатов или очень похожих фрагментов) и кросс-языкового поиска кода. Как отмечено в ряде исследований, клоны программного обеспечения наносят ущерб поддержке, развитию и сопровождению программного обеспечения (см., например, [2, 3]).

Рекомендательные системы в научной деятельности

Рекомендательные системы, наряду с системами поиска, являются наиболее распространенными системами в информационном пространстве. Базовой особенностью рекомендательных технологий является использование алгоритмов, способных учитывать предпочтения отдельного пользователя или категории пользователей в процессе создания персонализированных рекомендаций.

Имеется несколько определений рекомендательных систем, в каждом их которых сделан акцент на определенные их особенности (см., например, [7–9]). В работе [10] проведен анализ различных определений рекомендательных систем, даны ссылки на публикации и приведены сравнения определений с обоснованиями.

В основном в информационном пространстве используются рекомендательные системы, основанные на коллаборативной фильтрации. В научной деятельности применяют рекомендательные системы специального типа, в частности основанные на контенте с учетом семантики (Semantics-Aware Content-Based Recommender Systems) (см., например, [8]). К этому типу систем относится и рекомендательная система, представленная в настоящей работе.

Отметим работу [11], в которой предложена рекомендательная система поиска близких документов в физико-математическом контенте. Она основана на использовании онтологии профессиональной математики.

В работах [12, 13] создана рекомендательная система поиска экспертов для рецензирования математических работ в научном журнале. Эту систему авторы отнесли к типу рекомендательных систем конкретного случая (Case-Based Recommender Systems). Этот тип систем детально описан в [14], где представлены основные свойства таких систем и области их применения.

Рекомендательная система тематической классификации научных журналов предложена в [15]. Она основана на использовании рубрикаторов и классификаторов научно-технической информации, а также онтологии семантической библиотеки предметных областей SciLibRu.

Поскольку программный код в настоящее время рассматривается как самостоятельная единица научного знания, необходимы соответствующие программные инструменты, аналогичные созданным для управления научным контентом. Сегодня создаются исследовательские инфраструктуры (см., например, [16, 17]), цель которых – это интеграция и совместное использование научных документов, исследовательских данных и программ. Для исследовательских данных и программ разрабатываются инструменты поддержки их жизненного цикла, в том числе рекомендательные системы поиска семантически близких фрагментов программного кода.

Обзор близких по тематике исследований

Далее дан анализ существующих инструментальных средств и методов, направленных на решение задач анализа программного кода, выявления дубликатов и поиска функционально близких фрагментов в кросс-языковой среде.

Одним из первых значимых решений в области статического анализа межъязыковых клонов является инструмент LICCA, представленный в [2]. В его основе лежит использование обогащенных конкретных синтаксических деревьев (enriched Concrete Syntax Tree, eCST), которые позволяют унифицировать синтаксические конструкции различных языков программирования (C, Java, JavaScript, Modula-2 и Scheme) за счет введения универсальных узлов. Процесс сопоставления реализуется через сериализацию деревьев и применение модифицированного алгоритма поиска наибольшей общей подпоследовательности (Longest Common Subsequence, LCS). Несмотря на высокую точность на уровне синтаксических единиц, LICCA имеет существенные ограничения: система чувствительна к порядку следования инструкций и требует сопоставимой длины фрагментов кода, что затрудняет ее применение для детекции сложных семантических клонов.

Дальнейшее развитие подходов к синтаксическому анализу нашло отражение в модели, описанной в работе [3]. В отличие от инструментов, полагающихся на промежуточные представления, в этой модели производится анализ

сходства на основе 9 специфических синтаксических признаков, значения которых остаются относительно стабильными для функционально эквивалентного кода на различных языках. Архитектура системы включает «фильтр действий», основанный на семантическом сходстве вызовов API, которое вычисляется с помощью модели Word2Vec и анализа документации библиотек. Для классификации использована глубокая нейронная сеть с сиамской архитектурой, обучаемая на размеченных наборах данных для сопоставления признаков в едином векторе.

Метод COSAL (Code-to-Code Search Across Languages), ориентированный на кросс-языковой поиск, предлагает гибридный подход, объединяющий статический и динамический анализ без использования моделей машинного обучения [4]. Система оценивает релевантность кода по трем независимым критериям: сходство токенов, структурное сходство (на основе деревьев редактирования) и поведенческое сходство (анализ отношений ввода-вывода с использованием инструментария, представленного в [18]). Финальное ранжирование результатов осуществляется с помощью недоминируемой сортировки, что позволяет сбалансировать визуальное сходство кода с его фактическим функциональным поведением без потери нюансов, характерных для агрегированных метрик.

Переход к использованию предобученных моделей представления программного кода сопровождался появлением системы C4 (Contrastive Cross-Language Code Clone Detection) [5]. Этот метод базируется на архитектуре CodeBERT, которая трансформирует фрагменты кода в высокоразмерные векторные эмбеддинги. Ключевой особенностью C4 является применение контрастивного обучения (Contrastive Learning): модель обучается минимизировать расстояние между эмбеддингами функционально эквивалентных программ и максимизировать его для различных задач. Это позволяет эффективно выявлять семантические клоны 4-го типа, которые имеют идентичное поведение при полностью различном синтаксическом исполнении. Декларирована поддержка языков программирования Java, Python, C++ и C#.

Метод, представленный в [6], развивает идеи контрастивного подхода, адаптируя большие языковые модели для поиска кода за счет интеграции статических и динамических признаков на этапе обучения. В отличие от традиционных систем динамического анализа, в этом методе производится кодирование

информации о времени выполнения в виде оценки семантического сходства (Semantic Similarity Score, SSS) только в процессе тренировки модели, что избавляет от необходимости запускать код в момент выполнения поискового запроса. Кроме того, это первая система, использующая как положительные, так и отрицательные эталонные образцы в процессе дообучения, что значительно повышает точность поиска в условиях синтаксических различий между языками запроса и базы данных.

Новым направлением в области кросс-языкового обнаружении программных клонов стала модель, представленная в работе [19]. Эта модель предназначена для поиска клонов в условиях zero-shot-обучения, т. е. без использования параллельных межъязыковых наборов данных. С ее помощью предложено решение задачи выравнивания представлений через три механизма: контрастивное предсказание сниппетов (contrastive snippet prediction, CSP) для построения изоморфного векторного пространства, доменно-ориентированное обучение для устранения языковой специфики и обучение с циклической согласованностью (cycle consistency). Такой подход позволяет системе эффективно сопоставлять функции даже на тех языках программирования, которые не были представлены в обучающей выборке, обеспечивая высокую степень универсальности в открытых научных репозиториях.

Для высокоточного обнаружения межъязыковых клонов на семантическом уровне предложена модель FEGAT (Flow-Enhanced Graph Attention Network) [20]. Используемый в ней метод основан на графовых представлениях, обогащенных информацией о потоках данных и управления. Архитектура решения предполагает построение графа на основе абстрактного синтаксического дерева, дополненного ребрами потоков, которые затем поступают на вход предобученной модели CodeBERT для формирования первичных векторов узлов, насыщенных семантической информацией. Ключевым компонентом системы является нейронная сеть внимания на графах (Graph Attention Network, GAT), которая обучается извлекать компактные представления функциональной логики программ для последующего вычисления оценки их сходства.

Оценка сходства программных кодов Android-приложений представлена в работах [21, 22]. В них задача оценки сходства сведена к оценке сходства мно-

жеств графов потока управления. Значение сходства вычисляется на основе матрицы сходства. Графы потока управления сравниваются при помощи алгоритмов вычисления расстояний редактирования графов и расстояния Левенштейна [23]. Хоть семантика программного кода учитывается во всех приведенных исследованиях, в них поддерживаются до 4 из 20 языков из списка языков программирования, которые наиболее активно используются в настоящее время (см., например, [24, 25]).

Настоящая работа является продолжением исследований, представленных в [26]. Предложен метод кросс-языкового поиска семантически близких фрагментов кода, представленных на 19 языках программирования, входящих в текущий список наиболее активно используемых языков программирования: Bash, C, C#, C++, Go, Haskell, Java, JavaScript, Kotlin, Lua, PHP, Python, R, Ruby, Rust, Scala, Solidity, а также языков разметки CSS и HTML.

1. ПОСТАНОВКА ЗАДАЧИ

В работе используются термины *сходство*, *несходство* и *расстояние*. Эти термины понимаются в классическом смысле (см., например, [27]), а именно, под *сходством* объектов из множества O понимается функция $\sigma: O \times O \to R$, для которой выполнены условия *положительности*: $\forall x, y \in O, \sigma(x, y) \geq 0$, *максимальности*: $\forall x \in O, \forall y, z \in O, \sigma(x, x) \geq \sigma(y, z)$ и *симметричности*: $\forall x, y \in O, \sigma(x, y) = \sigma(y, x)$. Соответственно, двойственное понятие *несходства* объектов определяется как функция $\delta: O \times O \to R$, симметричная, положительная и удовлетворяющая условию *минимальности*: $\forall x \in O, \delta(x, x) = 0$. *Расстояние* на множестве объектов O определяется как функция несходства $\delta: O \times O \to R$, такая что выполнены условие *определенности*: $\forall x, y \in O, \delta(x, y) = 0$ тогда и только тогда, когда $x = y$, и *неравенство треугольника*: $\forall x, y, z \in O, \delta(x, y) + \delta(y, z) \geq \delta(x, z)$.

Обозначим через P множество всех возможных фрагментов программного кода; $DB \subset P$ – база данных программных фрагментов, доступная системе; $c \in P$ – входной программный фрагмент от пользователя; U – множество пользователей системы, где $u \in U$ – текущий пользователь, инициировавший запрос; $\theta \in [0,1]$ – предустановленное пороговое значение семантического сходства;

$\mathrm{sim}: P \times P \to [0,1]$ – функция, вычисляющая семантическое сходство двух программных фрагментов, значение 1 которой означает функциональную идентичность фрагментов, а 0 – полное отсутствие сходства.

Задача заключается в реализации рекомендательной системы, способной в соответствии с профилем пользователя и входным фрагментом кода сформировать набор персональных рекомендаций, состоящий из пар $R = \{(r_1, w_1), (r_2, w_2), \ldots, (r_k, w_k)\}$. Набор R упорядочен по убыванию значений второй компоненты, где $r_i \in DB$ – фрагмент кода из базы данных, $s_i = \mathrm{sim}(c, r_i) \geq \theta$ – сходство фрагмента кода из запроса и фрагмента из базы данных соответственно, значение которого не ниже установленного порога.

Ключевым требованием к системе является вычисление итогового релевантного веса w_i, который должен учитывать не только семантическое сходство s_i, но и контекст пользователя u. Таким образом, вес w_i является функцией трех параметров, а именно: базовой семантической близости s_i, $L_u(\mathrm{lang}(r_i))$ – значения коэффициента предпочтения языка программирования фрагмента r_i (учитывается в профиле пользователя по частоте взаимодействий); $I_u(r_i)$ – значения коэффициента соответствия предметной области фрагмента r_i научным предпочтениям текущего пользователя, извлеченным из его профиля ORCID.

Разрабатываемая система должна быть кросс-языковой, т. е. корректно оценивать функциональную эквивалентность, даже если фрагменты кода c и r_i представлены на различных языках программирования, а также обеспечивать поддержку фрагментов кода, написанных на 19 востребованных языках программирования и разметки из множества

L = {*Bash, C, C#, C++, CSS, Go, Haskell, HTML, Java, JavaScript, Kotlin, Lua, PHP, Python, R, Ruby, Rust, Scala, Solidity*}.

2. РЕКОМЕНДАТЕЛЬНЫЙ АЛГОРИТМ ПОИСКА БЛИЗКИХ ФРАГМЕНТОВ КОДА

Опишем алгоритм формирования рекомендаций по поиску близких фрагментов программного кода. На вход алгоритма подается запрос, содержащий фрагмент кода на любом из языков программирования из списка L. На выходе формируется набор рекомендаций, содержащий близкие фрагменты кода.

2.1. Общая схема алгоритма

Работа предложенного алгоритма начинается с этапа инициализации, на котором происходит подключение к базе данных DB программных фрагментов и задается пороговое значение θ сходства кодов, ограничивающее выдачу только наиболее релевантными решениями. При получении входного кода-запроса с система автоматически идентифицирует его язык программирования и выполняет процедуру векторизации для формирования эмбеддинга.

Дальнейший процесс организован в виде цикла, в ходе которого осуществляется последовательное сопоставление запроса с каждым фрагментом r из базы данных. Для каждого такого фрагмента также определяется язык программирования, после чего фрагмент преобразуется в векторное представление. На основе полученных векторов вычисляется значение косинусного сходства $\mathrm{sim}(c, r)$. Если полученное значение удовлетворяет установленному порогу θ, пара, состоящая из программного кода и его оценки сходства, включается в итоговое множество рекомендаций R.

Завершающим этапом являются сортировка сформированного набора в порядке убывания значения сходства и вывод ранжированного списка пользователю. Описанная последовательность действий описана в псевдокоде, представленном в Листинге 1.

Листинг 1. Псевдокод алгоритма поиска фрагментов программного кода, близких коду-запросу.

```
GET_SIMILAR_PROGRAMS(DB, c, threshold)
1   c_language ← detect_language(c)
2   c_embedding ← vectorize(c, c_language)
3   ▷ Инициализация пустого массива кортежей R
4   foreach r in DB
5       do r_language ← detect_language(r)
6          r_embedding ← vectorize(r, r_language)
7          s ← sim(c_embedding, r_embedding)
8          if s ≥ threshold
```

```
9                    then R ← R ∪ {(r, s)}
10  ▷ Сортировка R по убыванию второй компоненты (s)
11  return R
```

2.2. Создание набора фрагментов программного кода

Для реализации функционала потребовалось создать набор данных (датасет) фрагментов программного кода. Формирование экспериментального набора данных осуществлялось на базе крупномасштабного корпуса CodeNet, включающего более 13.9 млн программных образцов (порядка 500 млн строк кода) на 55 различных языках программирования [28].

Некоторые из рассматриваемых моделей векторизации (например, InferCode, предобученная на наибольшем количестве языков) для генерации эмбеддингов требуют обязательного предварительного указания языка программирования, на котором написан исходный код. В связи с этим решена задача предварительного определения языка программирования.

В рамках проведенного исследования из исходного корпуса было отобрано по 5000 репрезентативных фрагментов для каждого из целевых языков, поддерживаемых моделью векторизации InferCode. Из процесса выборки были исключены языки Solidity, R, HTML и CSS из-за отсутствия соответствующих данных в репозитории CodeNet. Сформированный набор данных опубликован на Zenodo и доступен для использования [29].

Датасет организован в виде системы папок, по одной на каждый язык программирования. Каждая папка содержит файлы с исходным кодом для соответствующего языка из множества L языков программирования, используемых в алгоритме. Названия файлов соответствуют идентификаторам решений из датасета CodeNet.

2.3. Метод определения языка программирования

Одной из задач при разработке системы является автоматическое определение языка программирования. Это связано с тем, что фрагменты кода, участвующие в сравнении, могут быть представлены на различных языках программирования. Кроме того, синтаксические конструкции фрагмента кода не всегда позволяют пользователю однозначно определить язык программирования.

Разработанный метод основан на анализе зарезервированных слов в языках программирования. Алгоритм определения языка программирования по содержимому кода состоит из следующих шагов. Работа алгоритма начинается с подключения к системе датасета зарезервированных слов для языков программирования из списка L. Для загруженного в систему фрагмента программного кода проводится предобработка, в частности удаление строковых литералов и комментариев, затем выполняется его токенизация. Далее запускается цикл по списку языков программирования, в ходе которого для каждого языка вычисляется количество вхождений его зарезервированных слов в списке токенов и рассчитывается отношение использованных зарезервированных слов к их общему количеству в загруженном фрагменте кода. В завершение производится сортировка списка языков по убыванию полученных значений, и наиболее вероятным считается тот язык программирования, который находится на первом месте в отсортированном списке. Описанный алгоритм и логика вычислений детально представлены в псевдокоде, приведенном в Листинге 2.

Листинг 2. Псевдокод алгоритма определения языка программного кода.

```
DETECT_LANGUAGE(RESERVED_WORDS, code)
1   tokens ← preprocessing(code)
2   ▷ Инициализация пустого ассоциативного массива scores с нулевыми
    значениями по умолчанию
3   foreach language in RESERVED_WORDS
4         do language_words ← RESERVED_WORDS[language]
5            counter ← 0
6            foreach keyword in language_words
7                  do if keyword in tokens
8                        then counter ← counter + 1
9            scores[language] ← counter / length[language_words]
10  ▷ Сортировка scores по убыванию
11  return scores[0]
```

Для обеспечения функционирования модуля идентификации языков из множества L был сформирован специализированный реестр зарезервированных слов, собранных на основе анализа официальной технической документации соответствующих языков программирования. Для задачи предобработки программного кода были разработаны и систематизированы формализованные шаблоны строковых литералов и комментариев, учитывающие синтаксические спецификации всех языков, заявленных в постановке задачи.

2.4. Векторное представление программного кода

Определение [30]. *Абстрактное синтаксическое дерево (АСД) – это иерархическая синтаксическая структура в виде конечного помеченного ориентированного дерева, представляющего программный код. Вершины этого дерева сопоставлены с операторами языка программирования, а листья – с передаваемыми в них операндами.*

На Рис. 1 в качестве примера приведен фрагмент кода алгоритма Евклида, а на Рис. 2 – представление этого кода в виде абстрактного синтаксического дерева.

```
1  def gcd(a: int, b: int):
2      while a != 0 and b != 0:
3          if a > b:
4              a = a % b
5          else:
6              b = b % a
7      return a + b
```

Рис. 1. Алгоритм Евклида поиска наибольшего общего делителя, представленный в виде кода на языке Python.

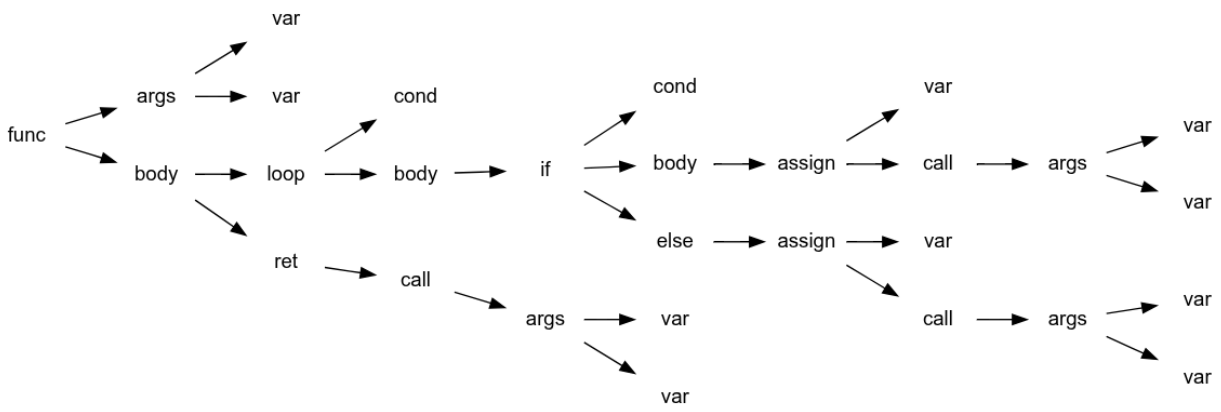


Рис. 2. Абстрактное синтаксическое дерево кода, представленного на Рис. 1.

Отметим, что АСД семантически однозначно представляют фрагменты кода на языках с гомоиконным синтаксисом. Анализ таких представлений представлен в [31].

Представление фрагментов кода в виде плотных векторов фиксированной размерности позволяет перевести задачу анализа программ в область вычислений расстояний в многомерных пространствах, где близость векторов определяет степень функционального или синтаксического сходства объектов. В современных исследованиях выделяется несколько ключевых подходов к векторизации, различающихся архитектурой нейронных сетей и типами используемых данных.

CodeBERT представляет собой бимодальную предобученную модель, предназначенную для захвата семантических связей между естественным языком (natural language, NL) и языками программирования (programming language, PL) [32]. Архитектура модели базируется на многослойном двунаправленном трансформере (Transformer). Обучение CodeBERT осуществляется с использованием гибридной функции потерь, включающей задачу маскированного языкового моделирования (masked language modeling, MLM) и оригинальную задачу детекции замененных токенов (replaced token detection, RTD). Использование RTD позволяет модели эффективно использовать как бимодальные данные (пары «код – документация»), так и большие массивы унимодального кода, что обеспечивает получение универсальных представлений, пригодных для задач поиска кода и генерации документации.

code2vec – это нейронная модель, ориентированная на обучение распределенным представлениям фрагментов кода на основе их синтаксической структуры [33]. Основная идея этого метода заключается в декомпозиции фрагмента кода на набор путей в его абстрактном синтаксическом дереве, соединяющих терминальные узлы. Каждый такой «путь» отображается в вектор, после чего сеть внимания (attention mechanism) вычисляет взвешенное среднее этих векторов для формирования итогового эмбеддинга кода фиксированной длины. Такой подход позволяет модели выделять синтаксические конструкции, наиболее значимые для семантики программы, что было успешно продемонстрировано на задаче предсказания имен методов.

InferCode реализует парадигму самообучения (self-supervised learning) для представления программного кода путем решения задачи предсказания поддеревьев в АСД [34]. В качестве кодировщика в системе используется древовидная сверточная нейронная сеть (tree-based convolutional neural network, TBCNN), которая напрямую обрабатывает иерархическую структуру дерева. Модель обучается предсказывать вероятность появления конкретных поддеревьев в заданном контексте АСД аналогично тому, как модель doc2vec предсказывает слова в документе. Ключевыми преимуществами InferCode являются его кросс-языковая универсальность (polyglot nature) и способность генерировать семантически богатые эмбеддинги, не привязанные к конкретной прикладной задаче.

GraphCodeBERT развивает идеи трансформерных архитектур, интегрируя в процесс обучения информацию о внутренней структуре программ [35]. В отличие от моделей, полагающихся только на токены или АСД, GraphCodeBERT использует графы потока данных, которые отражают семантические отношения зависимости между переменными. В предобучение включены две структурно-ориентированные задачи: предсказание наличия ребер графа потока данных и выравнивание переменных между текстом кода и узлами графа. Для эффективной обработки этих данных применяется специализированная функция маскированного внимания, управляемая графом (graph-guided masked attention).

UniXcoder представляет собой унифицированную кросс-модальную предобученную модель, поддерживающую широкий спектр задач: от классификации до авторегрессионной генерации и дополнения кода [36]. Эта модель использует матрицы маскированного внимания с префиксными адаптерами для гибкого переключения между режимами кодировщика и декодера. UniXcoder эффективно объединяет информацию из исходного кода, комментариев и АСД. Для параллельной обработки древовидных структур предложен метод взаимно однозначного отображения АСД в последовательность токенов, сохраняющий всю структурную информацию. Дополнительно применяются методы контрастивного обучения и кросс-модальной генерации для выравнивания представлений кода на различных языках программирования.

В настоящей работе для формирования векторного пространства программных фрагментов и оценки их семантического сходства в рамках рекомен-

дательной системы была выбрана модель InferCode. Выбор этой модели обусловлен ее способностью работать с широким набором языков программирования, приведенных в постановке задачи, и ориентацией на структурную семантику кода через анализ поддеревьев АСД.

3. РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ПОИСКА ПРОГРАММНОГО КОДА

В основе предлагаемого решения лежит парадигма рекомендательных систем, основанных на контенте (Content-Based Recommender Systems, CBRS), которые формируют персональные рекомендации, опираясь на описательные характеристики объектов и профили предпочтений пользователей. Основное допущение таких систем заключается в том, что интересы пользователя остаются стабильными во времени, поэтому ему предлагаются объекты, максимально схожие с теми, которые он положительно оценивал в прошлом. В контексте поиска программного кода это позволяет находить фрагменты, которые по своим функциональным и семантическим свойствам наиболее близки к запросу пользователя (см., например, [37]).

Пользователями настоящей рекомендательной системы являются исследователи и разработчики, работающие в мультидисциплинарных и мультиязычных проектах. Единицей рекомендации выступает фрагмент исходного кода из проиндексированной базы данных, функционально решающий ту же задачу, что и код-запрос, но потенциально написанный на другом языке программирования. Архитектура разработанной контентной рекомендательной системы (см. Рис. 3) включает три основных компонента: анализатор контента (`Content Analyzer`), модуль обучения профиля (`Profile Learner`) и компонент фильтрации и ранжирования (`Filtering Component`).

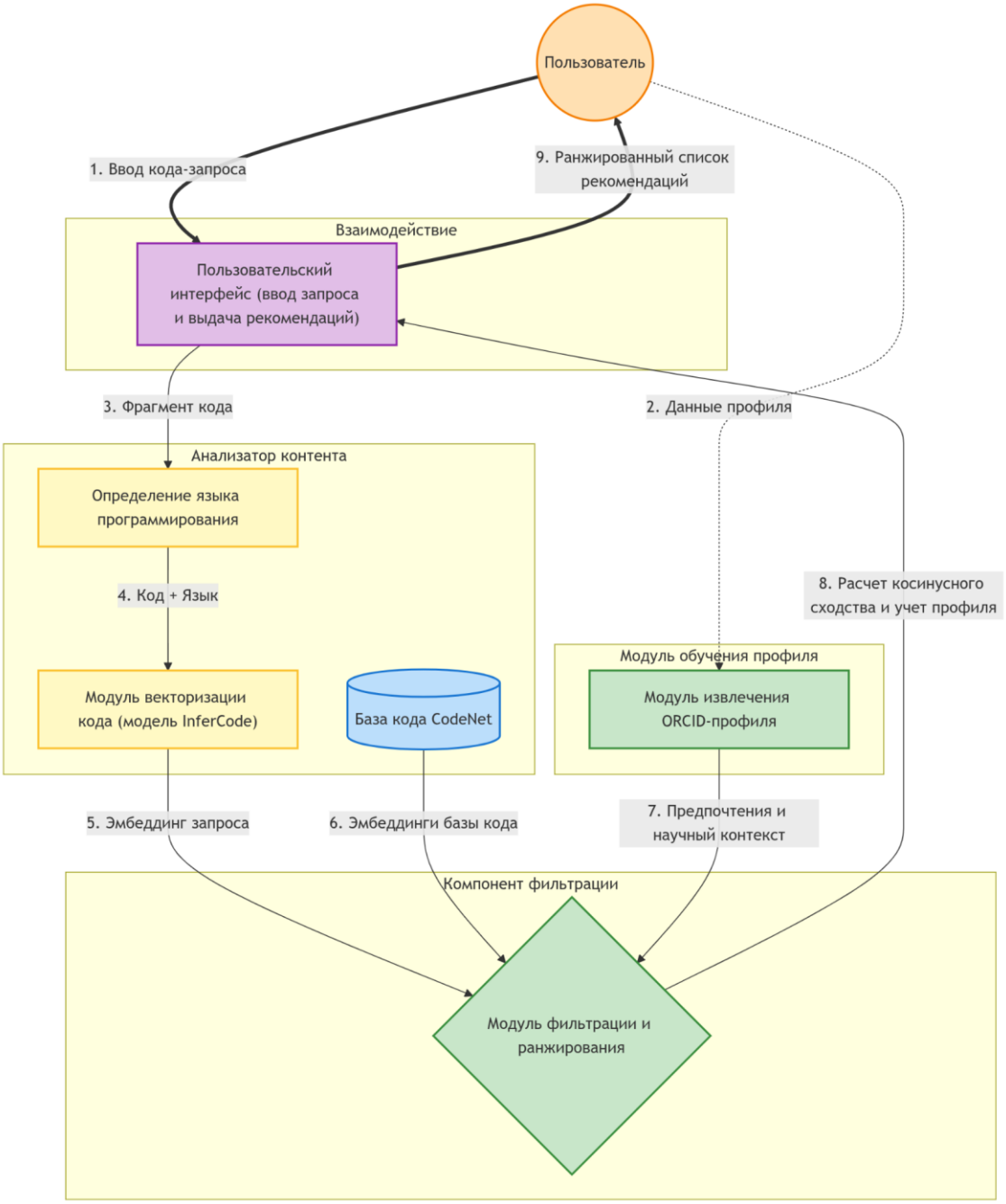


Рис. 3. Архитектура рекомендательной системы поиска близких фрагментов программного кода.

3.1. Анализатор контента

Анализатор контента отвечает за извлечение признаков из описаний объектов и создание структурированного представления, пригодного для машинной обработки. В рамках текущей версии системы этот процесс реализуется через построение абстрактных синтаксических деревьев и их последующую векто-

ризацию моделью InferCode. Для глубокого понимания семантики объектов используются методы дистрибутивной семантики (endogenous semantics), основанные на гипотезе о том, что слова (или элементы кода), встречающиеся в схожих контекстах, имеют схожие значения. Результатом работы таких моделей являются эмбеддинги – плотные векторы фиксированной размерности в многомерном пространстве, где расстояние между векторами служит мерой смысловой близости объектов. Базовой метрикой семантической релевантности между кодом-запросом c и фрагментом из базы данных r_i выступает косинусное сходство их эмбеддингов $s_i = \mathrm{sim}(c, r_i)$.

3.2. Модуль обучения профиля пользователя

Модуль обучения профиля собирает данные о предпочтениях пользователя, формируя модель его интересов на основе истории взаимодействий. В рамках этого модуля было принято допущение, по которому интересы пользователя имеют определенную стабильность, то есть ему предлагаются объекты, соответствующие технологическому стеку и предметной области его исследований. Профиль пользователя формируется на основе следующих двух факторов, относящихся к неявной и явной обратной связях соответственно.

1. Языковые предпочтения, т. е. технологический стек. Система автоматически фиксирует неявную обратную связь – факты взаимодействия с программным кодом на определенных языках программирования. Для каждого пользователя u и языка lang вычисляется нормализованная частота предпочтений $L_u(\mathrm{lang})$, показывающая долю взаимодействия с этим языком в общей истории пользователя.

2. Научный контекст, т. е. предметная область. Явная интеграция реализована через авторизацию по протоколу OAuth с использованием системы ORCID. При входе система извлекает из профиля исследователя набор ключевых слов, описывающих область его научных интересов. Это позволяет сформировать вектор интересов пользователя I_u для дополнительной контекстной фильтрации.

3.3. Компонент фильтрации и формирование рекомендаций

Главная задача компонента фильтрации состоит в преобразовании результатов базового алгоритма кросс-языкового поиска в персонализированную выдачу. Процесс формирования рекомендаций состоит из двух этапов.

На первом этапе производится фильтрация кандидатов. Из базы данных DB извлекается подмножество фрагментов кода $\{r_i\}$, для которых базовое семантическое сходство с запросом превышает установленный порог, т. е. $s_i \geq \theta$. Это гарантирует, что в рекомендации попадет только функционально релевантный код.

На втором этапе производится персонализированное ранжирование. Для каждого фрагмента-кандидата r_i вычисляется итоговый релевантный вес w_i, который определяет его позицию в итоговом списке рекомендаций. Вес рассчитывается как линейная комбинация базового контентного сходства и метрик профиля пользователя:

$$w_i = \alpha s_i + \beta L_u(\text{lang}(r_i)) + \gamma \text{Match}(I_u, \text{Metadata}(r_i)),$$

где s_i – семантическое сходство векторов кода-запроса и кандидата; $L_u(\text{lang}(r_i))$ – вес языкового предпочтения пользователя для языка, на котором написан кандидат r_i; $\text{Match}(I_u, \text{Metadata}(r_i))$ – функция оценки пересечения научных интересов пользователя (из ORCID) с метаданными или тегами контекста, привязанными к фрагменту кода в репозитории (например, принадлежность к математическим библиотекам, веб-разработке и т. д.); α, β, γ – настраиваемые гиперпараметры системы ($\alpha + \beta + \gamma = 1$), балансирующие вклады семантики и персонализации. Значения задаются эмпирически в конфигурации системы (по умолчанию наибольший вес α отдается семантическому сходству).

3.4. Пользовательский сценарий

Взаимодействие пользователя с системой происходит через веб-интерфейс. Пользователь вводит в систему фрагмент кода на любом из 19 поддерживаемых языков. Алгоритм автоматически определяет язык запроса, векторизует код и передает его в рекомендательное ядро.

В результате система возвращает пользователю список топ-N фрагментов кода, упорядоченный по убыванию веса w_i. Таким образом, на верхних позициях списка пользователь видит те фрагменты, которые не только максимально

точно реализуют исходный алгоритм, но и написаны на предпочтительных для него языках программирования и соответствуют профилю его исследований.

Применение контентного подхода обеспечивает системе ряд преимуществ: это независимость от данных других пользователей (рекомендации строятся только на основе интересов конкретного лица) и прозрачность, так как система может обосновать выбор результата наличием конкретных характеристик в коде [37].

4. ОЦЕНКА АЛГОРИТМА ФОРМИРОВАНИЯ РЕКОМЕНДАЦИЙ

Эффективность разработанного решения поиска сходных фрагментов программного кода и рекомендательной системы в целом оценивались с помощью серии вычислительных тестов. Осуществлялась проверка моделей векторизации на эталонном наборе данных, а также анализировалась точность разработанного вспомогательного алгоритма определения языка программирования.

4.1. Валидация моделей векторизации

Для вычисления показателей качества исследуемых моделей векторизации применялись набор данных и программная среда BigCloneEval [38], специально созданные для задачи оценки систем обнаружения программных клонов. В рамках валидации моделей метрика полноты (Recall) оценивалась для 8 различных типов клонов:

- тип-1 (Type-1) – точные копии (за исключением пробельных символов, форматирования и комментариев);
- тип-2 (Type-2) – объединение всех клонов 2-го типа;
- тип-2 несогласованный (Type-2 blind) – переименование переменных без сохранения соответствия;
- тип-2 согласованный (Type-2 consistent) – последовательное переименование переменных (один к одному);
- очень сильный тип-3 (Very-Strongly Type-3) – клоны с синтаксической схожестью в диапазоне [90, 100);
- сильный тип-3 (Strongly Type-3) – клоны с синтаксической схожестью в диапазоне [70, 90);
- умеренный тип-3 (Moderately Type-3) – клоны с синтаксической схожестью в диапазоне [50, 70);

- слабые тип-3/тип-4 (Weakly Type-3/Type-4) – клоны с синтаксической схожестью в диапазоне [0, 50).

В качестве критерия отсечения при подсчете метрик использовался статический порог косинусного сходства, равный 0.95. Результаты для базовых архитектур представлены в Табл. 1.

Табл. 1. Результаты валидации моделей для генерации эмбеддингов по программному коду.

Тип клона	CodeBERT	GraphCodeBERT	UniXcoder
Type-1	1.0000	0.9438	1.0000
Type-2	0.9495	0.5889	0.9495
Type-2 (blind)	0.9474	0.6132	0.9474
Type-2 (consistent)	0.9497	0.5867	0.9498
Very-Strongly Type-3	0.9596	0.6482	0.9596
Strongly Type-3	0.9507	0.3670	0.9508
Moderately Type-3	0.9901	0.0919	0.9904
Weakly Type-3 or Type-4	0.9997	0.0360	0.9998

Из таблицы видно, что модели UniXcoder и CodeBERT показывают сопоставимую и наиболее высокую эффективность. При этом UniXcoder демонстрирует незначительное преимущество на согласованном подтипе типа-2 и слабоструктурированных клонах сильного типа-3, умеренного типа-3 и слабых типах-3/4.

4.2. Валидация алгоритма определения языка программирования

Валидация алгоритма определения языка программирования производилась с использованием набора данных, описанного в разделе 2.2. Результаты сравнения предложенного алгоритма по ключевым словам с существующими аналогами представлены в Табл. 2. Разработанный нами алгоритм показал лучшие точность и скорость выполнения среди представленных решений при приемлемом значении полноты.

Табл. 2. Результаты валидации алгоритмов определения языка программирования.

Алгоритм	Accuracy	Precision	Recall	F1 score	Среднее время определения, мс
По ключевым словам	0.71	0.88	0.71	0.71	0.475
Guesslang	0.84	0.86	0.84	0.85	5.576
Pygments	0.03	0.19	0.03	0.03	20.724

ЗАКЛЮЧЕНИЕ

Предложена контентная рекомендательная система, использующая семантический анализ программного кода для кросс-языкового поиска семантически близких фрагментов в пространстве кодов. Разработанное решение акцентирует внимание на функциональной эквивалентности алгоритмов, позволяя находить смысловые аналоги на 19 различных языках программирования.

Основой анализатора контента выступает метод представления исходного кода в виде абстрактных синтаксических деревьев с последующей генерацией эмбеддингов на базе модели InferCode. Для обеспечения корректной работы конвейера обработки данных разработан алгоритм автоматической идентификации языка программирования по входному фрагменту, а также сформирован специализированный мультиязычный датасет на основе корпуса CodeNet.

В рамках рекомендательной системы был реализован модуль персонализации. Итоговое ранжирование рекомендаций осуществляется с учетом индивидуального профиля исследователя: система динамически взвешивает семантическое сходство кода с историей языковых предпочтений пользователя и областью его научных интересов, автоматически извлекаемых через авторизацию по профилю ORCID.

Так, предложенный подход, сочетающий векторизацию на основе синтаксических деревьев и метрическую оценку подобия, позволяет эффективно решать задачи интеллектуальной навигации в современных цифровых научных библиотеках. В перспективе разработанная рекомендательная система будет

интегрирована в исследовательскую инфраструктуру цифровой математической библиотеки Lobachevskii-DML [39].

Благодарности

Выражаем благодарность Наталии Павловне Тучковой за проявленный интерес к исследованию, значимые замечания и советы при оформлении статьи.

СПИСОК ЛИТЕРАТУРЫ

1. *Yang H., Nong Y., Wang S., Cai H.* Multi-Language Software Development: Issues, Challenges, and Solutions // IEEE Transactions on Software Engineering. 2024. Vol. 50, No. 3. P. 512–533. https://doi.org/10.1109/TSE.2024.3358258

2. *Vislavski T., Rakić G., Cardozo N., Budimac Z.* LICCA: A tool for cross-language clone detection // 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), Campobasso, Italy, 2018. P. 512–516. https://doi.org/10.1109/SANER.2018.8330250

3. *Nafi K.W., Kar T.S., Roy B., Roy C.K., Schneider K.A.* CLCDSA: Cross Language Code Clone Detection using Syntactical Features and API Documentation // 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, USA, 2019. P. 1026–1037. https://doi.org/10.1109/ASE.2019.00099

4. *Mathew G., Stolee K.T.* Cross-language code search using static and dynamic analyses // Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, New York, USA, 2021. P. 205–217. https://doi.org/10.1145/3468264.3468538

5. *Tao C., Zhan Q., Hu X., Xia X.* C4: contrastive cross-language code clone detection // ICPC '22: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension, New York, USA, 2022. P. 413–424. https://doi.org/10.1145/3524610.3527911

6. *Saieva A., Chakraborty S., Kaiser G.* Reinforest: Reinforcing Semantic Code Similarity for Cross-Lingual Code Search Models // 2024 IEEE International Conference on Source Code Analysis and Manipulation (SCAM). 2023. P. 177–188. https://doi.org/10.1109/SCAM63643.2024.00026

7. *Ricci F., Rokach L., Shapira B.* (Eds.) Recommender Systems Handbook. Springer New York, N.Y., 2022. 1060 p. https://doi.org/10.1007/978-1-0716-2197-4

8. *de Gemmis M., Lops P., Musto C., Narducci F., Semeraro G.* Semantics-

Aware Content-Based Recommender Systems // In: Ricci F., Rokach L., Shapira B. (Eds.) Recommender Systems Handbook. Springer, Boston, MA, 2015. P. 119–159. https://doi.org/10.1007/978-1-4899-7637-6_4

9. *Фальк К.* Рекомендательные системы на практике: практическое руководство. М.: ДМК Пресс, 2020. 256 с.

10. *Manouselis N., Drachsler H., Verbert K., Duval E.* Recommender Systems for Learning. Springer, 2013. https://doi.org/10.1007/978-1-4614-4361-2

11. *Elizarov A.M., Lipachev E.K., Zhizhchenko A.B., Zhil'tsov N.G., Kirillovich A.V.* Mathematical Knowledge Ontologies and Recommender Systems for Collections of Documents in Physics and Mathematics // Doklady Mathematics. 2016. Vol. 93, No. 2. P. 231–233. https://doi.org/10.1134/S1064562416020174

12. *Елизаров А.М., Липачев Е.К., Хайдаров Ш.М.* Метод автоматизированного подбора рецензентов научных статей, реализованный в информационной системе научного журнала // Научный сервис в сети Интернет: труды XXI Всерос. науч. конф. (23–28 сен. 2019, Новороссийск). М.: ИПМ им. М.В. Келдыша, 2019. С. 318–328. https://doi.org/10.20948/abrau-2019-94.
URL: http://keldysh.ru/abrau/2019/theses/94.pdf (дата доступа: 14.03.2026)

13. *Елизаров А.М., Липачев Е.К., Хайдаров Ш.М.* Рекомендательная система поиска экспертов для проведения научного рецензирования в математическом журнале // Электронные библиотеки. 2020. Т. 23, № 4. С. 708–732. https://doi.org/10.26907/1562-5419-2020-23-4-708-732

14. *Smyth B.* Case-based recommendation // In: Brusilovsky A., Kobsa W. (Eds.). The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer, Berlin, 2007. P. 342–376.

15. *Атаева О.М., Тучкова Н.П., Дегтев А.Г.* Рекомендательная система на основе обобщенного указателя журналов // Онтология проектирования. 2025. Т. 15, № 4. С. 598–613. https://doi.org/10.18287/2223-9537-2025-15-4-598-613

16. The MaRDI consortium. MaRDI: Mathematical Research Data Initiative Proposal. 2022. https://doi.org/10.5281/zenodo.6552436

17. *Kalinin N.A., Skvortsov N.A.* Difficulties of FAIR Principles Implementation in Cross-Domain Research Infrastructures // Lobachevskii J. Math. 2023. Vol. 44, No. 1. P. 147–156. https://doi.org/10.1134/S199508022301016X

18. *Mathew. G, Parnin C., Stolee K.T.* SLACC: simion-based language agnostic

code clones // Proc. of the ACM/IEEE 42nd International Conference on Software Engineering. 2020. P. 210–221. https://doi.org/10.1145/3377811.3380407

19. *Li J., Tao C., Jin Z., Liu F., Li J., Li G.* ZC^3: Zero-Shot Cross-Language Code Clone Detection // 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE). 2023. P. 875–887. https://doi.org/10.1109/ASE56229.2023.00210

20. *Hu M., Yang J., Zhou W.* Cross-language code clone detection via flow-enhanced graph attention network // The Computer Journal. 2026. https://doi.org/10.1093/comjnl/bxaf146

21. *Петров В.В.* Система автоматизации численной оценки сходства Android-приложений // Электронные библиотеки. 2024. Т. 27, № 3. С. 336–365. https://doi.org/10.26907/1562-5419-2024-27-3-336-365

22. *Petrov V.V.* Automated system for numerical similarity evaluation of android applications // Automatic documentation and mathematical linguistics. 2024. Vol. 58, No. 3. P. 131–142. https://doi.org/10.3103/S0005105525700207

23. *Riesen K.* Structural Pattern Recognition with Graph Edit Distance. Springer, Cham, 2015. 158 p. https://doi.org/10.1007/978-3-319-27252-8

24. The Top Programming Languages 2025 // IEEE Spectrum. 2025. URL: https://spectrum.ieee.org/top-programming-languages-2025 (дата доступа: 14.03.2026)

25. Топ языков программирования в 2025 году: рейтинг IEEE и влияние на него языковых моделей // Хабр-блог, 2025. URL: https://habr.com/ru/companies/selectel/articles/951348 (дата доступа: 14.03.2026)

26. *Зорин В.И., Липачев Е.К.* Метод вычисления меры сходства фрагментов программного кода // Системы высокой доступности. 2026. Т. 22, № 1. С. 47−50. https://doi.org/ 10.18127/j20729472-202601-09

27. *Euzenat J., Shvaiko P.* Basic similarity measures // In: Ontology Matching. Springer, Berlin, Heidelberg, 2013. P. 85–120. https://doi.org/10.1007/978-3-642-38721-0_5

28. *Puri R. et al.* CodeNet: A Large-Scale AI for Code Dataset for Learning a Diversity of Coding Tasks // NeurIPS Datasets and Benchmarks. 2021. https://doi.org/10.48550/arXiv.2105.12655

29. *Зорин В.И.* Programming Language Detection Dataset (1.0.0). https://doi.org/10.5281/zenodo.15661548

30. *Ахо А.В., Сети Р. Ульман Дж.Д.* Компиляторы: принципы, технологии и инструменты. М.: Вильямс, 2003. 768 с.

31. *Городняя Л.В.* Формы для показа результатов сравнения языков программирования на примере диалектов языка LISP // Электронные библиотеки. 2026. Т. 29 (1). С. 24–59. https://doi.org/10.26907/1562-5419-2026-29-1-24-59

32. *Feng Z., Guo D., Tang D., Duan N., Feng X., Gong M., Shou L., Qin B., Liu T., Jiang D., Zhou M.* CodeBERT: A Pre-Trained Model for Programming and Natural Languages // Empirical Methods in Natural Language Processing. 2020. P. 1536–1547. https://doi.org/10.18653/v1/2020.findings-emnlp.139

33. *Alon U., Zilberstein M., Levy O., Yahav E.* code2vec: learning distributed representations of code // Proc. of the ACM on Programming Languages. 2019. Vol. 3, No. POPL. P. 1–29. https://doi.org/10.1145/3290353

34. *Bui N.D.Q., Yu Y., Jiang L.* InferCode: Self-Supervised Learning of Code Representations by Predicting Subtrees // 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). 2020. P. 1186–1197. https://doi.org/10.1109/ICSE43902.2021.00109

35. *Guo D. et al.* GraphCodeBERT: Pre-training Code Representations with Data Flow // arXiv:2009.08366. 2020. https://doi.org/10.48550/arXiv.2009.08366

36. *Guo D., Lu S., Duan N., Wang Y., Zhou M., Yin J.* UniXcoder: Unified Cross-Modal Pre-training for Code Representation // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers), Dublin, 2022. P. 7212–7225. https://doi.org/10.18653/v1/2022.acl-long.499

37. *Musto C., Gemmis M.d.F., Lops P., Narducci F., Semeraro G.* Semantics and Content-Based Recommendations // F. Ricci, L. Rokach, B. Shapira (Eds.) Recommender Systems Handbook. Springer New York, N.Y., 2022. P. 251–298. https://doi.org/10.1007/978-1-0716-2197-4_7

38. *Svajlenko J., Roy C.K.* BigCloneEval: A Clone Detection Tool Evaluation Framework with BigCloneBench // 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME). 2016. P. 596–600. https://doi.org/10.1109/ICSME.2016.62

39. *Елизаров А.М., Кириллович А.В., Липачев Е.К., Невзорова О.А.* Цифровая экосистема OntoMath как подход к построению пространства математических знаний // Электронные библиотеки. 2023. Т. 26. № 2. С. 154–202. https://doi.org/10.26907/1562-5419-2023-26-2-154-202

A RECOMMENDATION SYSTEM FOR FINDING SEMANTICALLY SIMILAR FRAGMENTS OF PROGRAM CODE

V. I. Zorin[1] [0009-0004-0271-1882], E. K. Lipachev[2] [0000-0001-7789-2332]

[1]*Kazan National Research Technical University named after A. N. Tupolev — KAI, Kazan, Russia*

[2]*Kazan Federal University, Kazan, Russia*

[2]*Innopolis University, Innopolis, Russia*

[1]addefan@mail.ru, [2]elipachev@gmail.com

Abstract

Recommendation systems in the scientific information space serve as essential tools for search and navigation when working with scientific documents. Software code is currently considered as an object of scientific knowledge and, as a result, an important task is to create software lifecycle support systems, in particular, to find similar software solutions, detect code borrowings, analyze and evaluate code quality.

This paper proposes a content-based recommender system that provides users with a personalized list of code fragments that are functionally equivalent to the input query code presented in one of the programming languages from the established set.

The basic algorithm of the system is based on the representation of the program code in the form of an abstract syntax tree followed by the construction of a vector space of program codes. The semantic similarity of program codes is determined by the distance between code vectors in a multidimensional space.

The personalization of recommendations is achieved through a filtering module that ranks the retrieved fragments taking into account the user's profile. The factors under consideration are the language preferences of the user and his areas of scientific interests, extracted through integration with ORCID.

To ensure the system's operation, a specialized dataset was created based on

the CodeNet corpus. The problem of automated language detection from a snippet of the presented code in one of the 19 languages included in the current rating list of programming languages has also been solved.

Keywords*: abstract syntax tree, code embedding, content-based filtering, cross-language clone, cross-language code search, code similarity, recommender system.*

REFERENCES

1. *Yang H., Nong Y., Wang S., Cai H.* Multi-Language Software Development: Issues, Challenges, and Solutions // IEEE Transactions on Software Engineering. 2024. Vol. 50, No. 3. P. 512–533. https://doi.org/10.1109/TSE.2024.3358258

2. *Vislavski T., Rakić G., Cardozo N., Budimac Z.* LICCA: A tool for cross-language clone detection // 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER), Campobasso, Italy, 2018. P. 512–516. https://doi.org/10.1109/SANER.2018.8330250

3. *Nafi K.W., Kar T.S., Roy B., Roy C.K., Schneider K.A.* CLCDSA: Cross Language Code Clone Detection using Syntactical Features and API Documentation // 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, USA, 2019. P. 1026–1037. https://doi.org/10.1109/ASE.2019.00099

4. *Mathew G., Stolee K.T.* Cross-language code search using static and dynamic analyses // Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, New York, USA, 2021. P. 205–217. https://doi.org/10.1145/3468264.3468538

5. *Tao C., Zhan Q., Hu X., Xia X.* C4: contrastive cross-language code clone detection // ICPC '22: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension, New York, USA, 2022. P. 413–424. https://doi.org/10.1145/3524610.3527911

6. *Saieva A., Chakraborty S., Kaiser G.* Reinforest: Reinforcing Semantic Code Similarity for Cross-Lingual Code Search Models // 2024 IEEE International Conference on Source Code Analysis and Manipulation (SCAM). 2023. P. 177–188. https://doi.org/10.1109/SCAM63643.2024.00026

7. *Ricci F., Rokach L., Shapira B.* (Eds.) Recommender Systems Handbook. Springer New York, N.Y., 2022. 1060 p. https://doi.org/10.1007/978-1-0716-2197-4

8. *de Gemmis M., Lops P., Musto C., Narducci F., Semeraro G.* Semantics-Aware Content-Based Recommender Systems // In: Ricci F., Rokach L., Shapira B.

(Eds.) Recommender Systems Handbook. Springer, Boston, MA, 2015. P. 119–159. https://doi.org/10.1007/978-1-4899-7637-6_4

9. *Falk K.* Recommender Systems in Practice: A Practical Guide. Springer, Berlin, 2016. 340 p.

10. *Manouselis N., Drachsler H., Verbert K., Duval E.* Recommender Systems for Learning. Springer, 2013. https://doi.org/10.1007/978-1-4614-4361-2

11. *Elizarov A.M., Lipachev E.K., Zhizhchenko A.B., Zhil'tsov N.G., Kirillovich A.V.* Mathematical Knowledge Ontologies and Recommender Systems for Collections of Documents in Physics and Mathematics // Doklady Mathematics. 2016. Vol. 93, No. 2. P. 231–233. https://doi.org/10.1134/S1064562416020174

12. *Elizarov A.M., Lipachev E.K., Khaydarov S.M.* Method of automated selection of reviewers of scientific articles, implemented in the scientific journal information system // Nauchny`j servis v seti Internet. M: IPM im. Keldysha, 2019. P. 318–328. https://doi.org/10.20948/abrau-2019-94

13. *Elizarov A.M., Lipachev E.K., Khaydarov S.M.* Recommender system in the process of scientific peer review in mathematical journal // Russian Digital Libraries Journal. 2020. Vol. 23, No. 4. P. 708–732. https://doi.org/10.26907/1562-5419-2020-23-4-708-732

14. *Smyth B.* Case-based recommendation // In: Brusilovsky A., Kobsa W. (Eds). The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer, Berlin, 2007. P. 342–376.

15. *Ataeva O.M., Tuchkova N.P., Degtev A.G.* Recommendation system based on a generalized journal index // Ontology of Designing. 2025. Vol. 15, No. 4. P. 598–613. https://doi.org/10.18287/2223-9537-2025-15-4-598-613

16. The MaRDI consortium. MaRDI: Mathematical Research Data Initiative Proposal. 2022. https://doi.org/10.5281/zenodo.6552436

17. *Kalinin N.A., Skvortsov N.A.* Difficulties of FAIR Principles Implementation in Cross-Domain Research Infrastructures // Lobachevskii J. Math. 2023. Vol. 44, No. 1. P. 147–156. https://doi.org/10.1134/S199508022301016X

18. *Mathew. G, Parnin C., Stolee K.T.* SLACC: simion-based language agnostic code clones // Proc. of the ACM/IEEE 42nd International Conference on Software Engineering. 2020. P. 210–221. https://doi.org/10.1145/3377811.3380407

19. *Li J., Tao C., Jin Z., Liu F., Li J., Li G.* ZC3: Zero-Shot Cross-Language Code

Clone Detection // 2023 38th IEEE/ACM International Conference on Automated Software Engineering (ASE). 2023. P. 875–887.
https://doi.org/10.1109/ASE56229.2023.00210

20. *Hu M., Yang J., Zhou W.* Cross-language code clone detection via flow-enhanced graph attention network // The Computer Journal. 2026.
https://doi.org/10.1093/comjnl/bxaf146

21. *Petrov V.V.* Automated system for numerical similarity evaluation of android applications // Russian Digital Libraries Journal. 2024. Vol. 27, No. 3. P. 336–365. https://doi.org/10.26907/1562-5419-2024-27-3-336-365

22. *Petrov V.V.* Automated system for numerical similarity evaluation of android applications // Automatic documentation and mathematical linguistics. 2024. Vol. 58, No. 3. P. 131–142. https://doi.org/10.3103/S0005105525700207

23. *Riesen K.* Structural Pattern Recognition with Graph Edit Distance. Springer, Cham, 2015. 158 p. https://doi.org/10.1007/978-3-319-27252-8

24. The Top Programming Languages 2025. 2025.
URL: https://spectrum.ieee.org/top-programming-languages-2025 (Accessed: 22.03.2026).

25. Top yazy`kov programmirovaniya v 2025 godu: rejting IEEE i vliyanie na nego yazy`kovy`x modelej // Habr-blog, 2025. URL: https://habr.com/ru/companies/selectel/articles/951348 (Accessed: 22.03.2026).

26. *Zorin V.I., Lipachev E.K.* A method for calculating the similarity measure of program code fragments // Highly Available Systems. 2026. Vol. 22, No. 1. P. 47−50. https://doi.org/10.18127/j20729472-202601-09

27. *Euzenat J., Shvaiko P.* Basic similarity measures // In: Ontology Matching. Springer, Berlin, Heidelberg, 2013. P. 85–120.
https://doi.org/10.1007/978-3-642-38721-0_5

28. Puri R. et al. CodeNet: A Large-Scale AI for Code Dataset for Learning a Diversity of Coding Tasks // NeurIPS Datasets and Benchmarks. 2021.
https://doi.org/10.48550/arXiv.2105.12655

29. Zorin V.I. Programming Language Detection Dataset (1.0.0).
https://doi.org/10.5281/zenodo.15661548

30. *Aho A.V., Lam M.S., Sethi R., Ullman J.D.* Compilers: Principles, Techniques, and Tools (2 ed.). Addison-Wesley, Boston, 2006. 1006 p.

31. *Gorodnyaya L.V.* Forms for displaying the results of comparison of programming languages using the example of dialects of the LISP language // Russian Digital Libraries Journal. 2026. Vol. 29, No. 1. P. 24–59. https://doi.org/10.26907/1562-5419-2026-29-1-24-59

32. *Feng Z., Guo D., Tang D., Duan N., Feng X., Gong M., Shou L., Qin B., Liu T., Jiang D., Zhou M.* CodeBERT: A Pre-Trained Model for Programming and Natural Languages // Empirical Methods in Natural Language Processing. 2020. P. 1536–1547. https://doi.org/10.18653/v1/2020.findings-emnlp.139

33. *Alon U., Zilberstein M., Levy O., Yahav E.* code2vec: learning distributed representations of code // Proc. of the ACM on Programming Languages. 2019. Vol. 3, No. POPL. P. 1–29. https://doi.org/10.1145/3290353

34. *Bui N.D.Q., Yu Y., Jiang L.* InferCode: Self-Supervised Learning of Code Representations by Predicting Subtrees // 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). 2020. P. 1186–1197. https://doi.org/10.1109/ICSE43902.2021.00109

35. *Guo D. et al.* GraphCodeBERT: Pre-training Code Representations with Data Flow // arXiv:2009.08366. 2020. https://doi.org/10.48550/arXiv.2009.08366

36. *Guo D., Lu S., Duan N., Wang Y., Zhou M., Yin J.* UniXcoder: Unified Cross-Modal Pre-training for Code Representation // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, 2022. P. 7212–7225. https://doi.org/10.18653/v1/2022.acl-long.499

37. *Musto C., Gemmis M.d., Lops P., Narducci F., Semeraro G.* Semantics and Content-Based Recommendations // F. Ricci, L. Rokach, B. Shapira (Eds.) Recommender Systems Handbook. Springer New York, N.Y., 2022. P. 251–298. https://doi.org/10.1007/978-1-0716-2197-4_7

38. *Svajlenko J., Roy C.K.* BigCloneEval: A Clone Detection Tool Evaluation Framework with BigCloneBench // ICSME, 2016. P. 596–600. https://doi.org/10.1109/ICSME.2016.62

39. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A.* Digital Ecosystem OntoMath as an Approach to Building the Space of Mathematical Knowledge // Russian Digital Libraries Journal. 2023. Vol. 26, No. 2. P. 154–202. https://doi.org/10.26907/1562-5419-2023-26-2-154-202

СВЕДЕНИЯ ОБ АВТОРАХ

ЗОРИН Виталий Иванович – магистрант Института компьютерных технологий и защиты информации Казанского национального исследовательского технического университета им. А.Н. Туполева – КАИ. Научные интересы: программная инженерия, обработка естественного языка, рекомендательные системы.

Vitaly Ivanovich ZORIN – graduate student at the Institute of Computer Technology and Information Security of Kazan National Research Technical University named after A.N. Tupolev – KAI. Research interests: software engineering, natural language processing, recommendation systems.

email: addefan@mail.ru

ORCID: 0009-0004-0271-1882

ЛИПАЧЕВ Евгений Константинович – кандидат физико-математических наук, доцент кафедры цифровой аналитики и технологий искусственного интеллекта Института информационных технологий и интеллектуальных систем Казанского федерального университета, доцент Университета Иннополис. Научные интересы: цифровые библиотеки, интеллектуальный анализ данных, рекомендательные системы, технологии извлечения знаний.

Evgeny Konstantinovich LIPACHEV – Candidate of Physics and Mathematics, Associate Professor, Kazan Federal University, Innopolis University. Research interests: digital libraries, data mining, recommender systems, knowledge extraction technologies.

email: elipachev@gmail.com

ORCID: 0000-0001-7789-2332

Материал поступил в редакцию 14 марта 2025 года

УДК 004

ФОРМИРОВАНИЕ И РАЗМЕТКА КОРПУСА РУССКОЯЗЫЧНЫХ НОВОСТНЫХ ТЕКСТОВ ДЛЯ АВТОМАТИЗИРОВАННОГО ВЫЯВЛЕНИЯ ПОЛИТИЧЕСКИХ МАНИПУЛЯЦИЙ

Н. Л. Кулюлина[0009-0006-1715-1114]

Московский физико-технический институт, г. Долгопрудный, Россия

kuliulina.nl@phystech.edu

Аннотация

Исследована проблема создания специализированных корпусных ресурсов для задач автоматизированного анализа политических манипуляций в русскоязычных текстах. Несмотря на активное развитие методов семантического и вычислительного анализа текстов, существующие корпусные ресурсы и схемы разметки в основном ориентированы на англоязычные данные и плохо учитывают языковую и контекстуальную специфику русскоязычных новостных средств массовой информации (СМИ).

Целями исследования были создание специализированного корпуса русскоязычных новостных текстов и разработка схемы разметки, ориентированной на автоматизированный анализ политических манипуляций с учетом особенностей русскоязычного медиапространства.

В рамках проведенного исследования сформирован корпус фраз, извлеченных из русскоязычных новостных текстов и опубликованных в период 2010–2019 гг., и разработана схема разметки манипулятивных техник. В основе разметки лежит адаптация международных классификаций манипулятивных стратегий, сведенных к ограниченному числу интерпретируемых техник, релевантных для анализа русскоязычных новостных текстов. Предлагаемая схема охватывает эмоциональные, аргументативные и контекстуальные формы манипулятивного воздействия.

Полученные корпус и схема разметки могут использоваться в качестве эмпирической основы для разработки и тестирования методов автоматизированного анализа политических манипуляций в русскоязычных новостных СМИ, а также дальнейших исследований политических и медиа-текстов.

Ключевые слова: *медиа-манипуляции, русскоязычные СМИ, корпус текстов, разметка данных, манипулятивные техники, политическая коммуникация, семантический анализ, вычислительный дискурс-анализ.*

ВВЕДЕНИЕ

В современных новостных медиа политические манипуляции нередко реализуются не через прямое искажение фактов, а через контекстуальные механизмы отбора тем, расстановки акцентов и подбора терминологии, формирующие интерпретацию политических событий при сохранении видимости нейтрального информирования [1, 2]. Такие механизмы проявляются в устойчивых способах текстового и контекстуального оформления сообщений и не сводятся к отдельным лексическим маркерам, что делает их анализ центральной задачей исследований политической коммуникации [3, 4].

Эмпирическое изучение политических манипуляций требует сопоставления теоретических подходов с наблюдаемыми единицами текста, которые могут быть систематически выделены и описаны. Традиционные качественные методы контент- и дискурс-анализа обеспечивают такую связь теории и эмпирики, однако их высокая трудоемкость и ограниченная масштабируемость существенно затрудняют анализ больших массивов новостных данных. При этом в условиях цифровизации медиапространства при отсутствии специализированных корпусов новостных текстов с единой схемой разметки манипулятивных приемов применение автоматизированных методов анализа политических манипуляций также оказывается затруднительным [5].

Несмотря на развитие корпусных ресурсов и вычислительных методов анализа, направленных на выявление манипулятивных стратегий в новостных текстах, большинство существующих решений ориентировано на англоязычные данные и в ограниченной степени применимо к русскоязычным новостным текстам [6, 7]. Недостаток специализированных корпусов и схем разметки, учитывающих языковую и контекстуальную специфику русскоязычных СМИ, существенно ограничивает возможности автоматизированного анализа политических манипуляций [8, 9].

В связи с этим целью настоящей статьи является представление корпуса

фраз, извлеченных из русскоязычных новостных текстов, и схемы разметки манипулятивных техник, разработанной с учетом особенностей русскоязычных новостных текстов. Корпус и схема разметки рассматриваются как эмпирическая основа для решения задачи автоматизированного выявления манипулятивных техник в русскоязычных новостных текстах на уровне предложения. Задача сформулирована как двухэтапная: на первом этапе осуществляется бинарная классификация предложений на манипулятивные и неманипулятивные, а на втором этапе — классификация манипулятивных предложений по типам используемых техник.

ОБЗОР БЛИЗКИХ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЙ

Одним из наиболее известных корпусных ресурсов в рамках задачи выявления политических манипуляций является англоязычный Media Frames Corpus, предложенный Д. Кардом и соавторами [6], в котором новостные тексты размечены по тематическим категориям, описывающим типичные способы представления и объяснения одних и тех же событий в новостных медиа. Этот корпус заложил основу для последующих исследований, демонстрирующих, что наличие качественной разметки позволяет выявлять манипулятивные стратегии в больших массивах новостных данных и отслеживать их динамику во времени и по источникам. В дальнейшем подход был расширен за счет интеграции метаданных документов, что повысило точность и интерпретируемость автоматизированного анализа [10].

Отдельное направление исследований связано с выявлением манипулятивных техник на уровне фрагментов текста. В работах Дж. Да Сан Мартино и соавторов была предложена типология из 18 манипулятивных техник и создан корпус новостных текстов с разметкой на уровне фраз [7, 11]. Эти исследования показали, что переход от документного уровня анализа к более мелким текстовым единицам позволяет точнее зафиксировать конкретные приемы манипулятивного воздействия, в том числе с использованием автоматизированных методов анализа. При этом сами авторы подчеркивают, что устойчивость моделей в значительной степени определяется качеством и репрезентативностью разметки.

Применение аналогичных методологий к русскоязычным новостным тек-

стам остается ограниченным. Существующие исследования показали, что модели и схемы разметки, разработанные для английского языка, плохо переносятся на русскоязычные тексты из-за лингвистических различий (например, в русском языке активно используются эвфемизмы, конструкты двойного значения, грамматические конструкции с оценочной окраской) [9], а также из-за особенностей медиасреды (различные медиа используют принципиально разные коммуникативные стратегии) [8]. Отсутствие открытых и специализированных размеченных корпусов для анализа политических манипуляций в русскоязычных новостных СМИ существенно сдерживает развитие воспроизводимых и масштабируемых исследований в данной области.

МЕТОДЫ

В рамках проведенного исследования был сформирован корпус фраз на материале русскоязычных новостных публикаций, находящихся в открытом доступе в Интернете. В корпус включены тексты, опубликованные в период с 2010 по 2019 г., что позволяет снизить влияние краткосрочных информационных кампаний и отдельных политических событий на структуру данных и соответствует практике корпусных исследований медиа [6, 8]. Источниками данных выступили русскоязычные новостные СМИ, различающиеся по редакционным стратегиям и позиционированию: *Газета.Ru, Лента.ру, RT (Россия сегодня)* и др., что позволяет учитывать вариативность новостных текстов и потенциально способствует устойчивости моделей к источниковому сдвигу.

Масштаб корпуса делает невозможной полную ручную разметку текстов, особенно при переходе к фразовому уровню анализа, где количество наблюдений возрастает кратно. В связи с этим в работе использована стратегия ограниченной экспертной разметки репрезентативной выборки единиц с возможностью последующего расширения данных с помощью генерации синтетических данных.

В качестве единицы анализа в настоящем исследовании использован фразовый фрагмент (предложение), извлеченный из новостного текста. Выбор фразового уровня анализа обусловлен спецификой манипулятивных техник: большинство из них реализуется на уровне отдельных высказываний через подбор

слов и структуру формирования конкретных предложений, которые задают аргументативные конструкции и контекстуальные акценты [7]. Анализ на уровне целых документов, напротив, нередко затрудняет выявление конкретных приемов воздействия и приводит к смешению различных стратегий в рамках одного наблюдения.

Сегментация текстов на предложения осуществлялась автоматически с применением стандартных правил пунктуационной сегментации. Дополнительная фрагментация предложений не проводилась, поскольку большинство манипулятивных техник реализуется на уровне завершенного высказывания, а сохранение целостности предложения повышает интерпретируемость разметки. Для каждой фразовой единицы в корпусе сохраняется информация об источнике, дате публикации, заголовке статьи и полной ссылке на исходный материал, что позволяет при необходимости восстанавливать расширенный контекст и анализировать пограничные случаи.

Схема разметки корпуса основана на определении понятия политической манипуляции как *формы скрытого информационного воздействия, реализуемого через конкретные языковые и контекстуальные техники при сохранении видимости нейтрального информирования* [14].

В качестве базовой типологии манипулятивных техник использована международная классификация SemEval [11], разработанная для задач автоматизированного выявления манипуляций в новостных текстах. При адаптации этой типологии к русскоязычным текстам отбор техник осуществлялся на основе трех критериев: (1) регулярность и воспроизводимость реализации техники на фразовом уровне, (2) интерпретируемость разметки без привлечения широкого контекстуального знания и (3) применимость техники к разнородным новостным тематикам и источникам. В результате исходный набор из 18 техник был редуцирован до 6 основных техник, релевантных для последующего корпусного и автоматизированного анализа. Выделенные техники соотносятся с тремя укрупненными типами манипулятивного воздействия: эмоциональным, аргументативным и контекстуальным. Итоговая схема включает следующие манипулятивные техники:

- **эмоционально заряженный язык** (Loaded language) – использование лексики с выраженной эмоциональной коннотацией;

- **навешивание ярлыков** (Name calling / labeling) – использование оценочных «ярлыков» по отношению к акторам или группам;
- **запугивание** (Appeal to fear / prejudice) – апелляция к страхам, угрозам или социальным предубеждениям;
- **апелляция к авторитету** (Appeal to authority) – ссылка на авторитет в качестве аргумента без критической проверки;
- **ложная причинность** (Causal oversimplification) – чрезмерное упрощение причинно-следственных связей;
- **ложная дихотомия** (Black-and-white fallacy) – представление ситуации в бинарной оппозиции.

Для формирования выборки предложений, подлежащих ручной разметке, применялся автоматический предварительный отбор кандидатов на основе эвристического анализа текста предложения без использования внешнего контекста. В качестве признаков потенциальной манипулятивности учитывались лексико-дискурсивные маркеры (апелляции к источникам и авторитетам, оценочная и эмоционально окрашенная лексика, обобщающие формулы), аргументативные конструкции (причинно-следственные связки, бинарные противопоставления), а также апелляции к угрозам и негативным последствиям. По совокупности этих признаков предложения ранжировались, после этого итоговая выборка для ручной разметки формировалась как комбинация предложений с высокой оценкой по итогам ранжирования и случайно отобранных нейтральных фрагментов.

Табл. 1. Распределение классов в размеченной выборке.

Класс	Число предложений	Доля
Бинарная классификация		
Манипулятивные	358	44.7%
Нейтральные	442	55.3%
Классификация по типам манипулятивных техник		
Эмоционально заряженный язык	94	26.3%
Апелляция к авторитету	70	19.6%
Навешивание ярлыков	69	19.3%

Ложная причинность	45	12.5%
Запугивание	43	12.0%
Ложная дихотомия	37	10.3%

Ручная разметка предложений осуществлялась экспертно с использованием разработанной схемы манипулятивных техник. В размеченной выборке наблюдался умеренный дисбаланс между манипулятивными и неманипулятивными предложениями (см. табл. 1); данное распределение не требует балансировки на уровне бинарной классификации. В то же время анализ распределения манипулятивных техник в размеченной выборке показал выраженный дисбаланс между классами.

Исходя из полученного распределения, можно утверждать, что балансировка данных на уровне бинарной классификации не является критически необходимой, тогда как для задачи классификации по типам манипулятивных техник требуется дополнительная работа с редкими классами. Реализация этого этапа планируется посредством дополнительного целевого отбора кандидатов из неразмеченного корпуса с последующей ручной валидацией, только после этого возможно использование данной разметки в качестве основы для генерации синтетических данных; качество и корректность синтетически сгенерированных данных предлагается оценивать посредством выборочной экспертной валидации, включающей проверку соответствия разметки исходной схеме манипулятивных техник, а также анализ типичных ошибок и пограничных случаев.

Кроме того, для осуществления предварительных экспериментов в рамках бинарной классификации был реализован этап генерации синтетических данных. Синтетические примеры были сформированы на основе исходных вручную размеченных предложений с использованием слабых преобразований текста, сохраняющих метку. В итоговом наборе данных (см. табл. 2) оригинальные и синтетические примеры хранились совместно и сопровождались явной маркировкой, указывающей на их происхождение.

Табл. 2. Распределение классов в размеченной выборке
(с дополнением синтетическими данными).

Класс\тип данных	Оригинальные	Синтетические	Итого
Манипулятивные	358	558	916 (44.2%)
Нейтральные	442	713	1155 (55.8%)
Итого	800	1271	2071

ПРИМЕРЫ И ЭКСПЕРИМЕНТЫ

Для демонстрации логики предложенной схемы разметки приведем иллюстративные примеры фраз из корпуса с указанием выявленных манипулятивных техник. Примеры отобраны из русскоязычных новостных публикаций Газета.Ru за 2010 год и отражают различные типы манипулятивного воздействия на фразовом уровне.

Пример 1 (Ложная дихотомия): *«Проблемы порнобизнеса ничем не отличаются от проблем любого другого бизнеса».* В данном фрагменте сложное и социально нагруженное явление представлено в виде упрощенной бинарной аналогии. Формулировка исключает альтернативные интерпретации и нивелирует специфические этические и правовые аспекты рассматриваемого феномена, тем самым формируя однозначную интерпретационную рамку.

Пример 2 (Апелляция к авторитету): *«Так, координатор государственно-патриотического движения заявил, что происходящее является прямым следствием внешнего давления».* В данном случае манипулятивное воздействие реализовано через апелляцию к неконкретизированному институциональному авторитету. Ссылка на позицию представителя организации использована как аргумент, не сопровождаемый независимым обоснованием или альтернативными точками зрения.

Пример 3 (Запугивание): *«Эксперты предупреждают, что дальнейшее развитие ситуации может привести к резкому росту преступности и дестабилизации обстановки в стране».* В данном фрагменте использована апелляция к неопределенной, но социально значимой в российском контексте угрозе (ассоциация с кризисом 90-х годов XX в.). Формулировка апеллирует к страху перед негативными последствиями, не предоставляя конкретных механизмов или

проверяемых оснований для подобного прогноза, что способствует формированию тревожной интерпретационной рамки у читателя.

Пример 4 (неманипулятивный фрагмент): *«Будущие молодожены прибыли к собору по отдельности: 32-летний Вестлинг приехал вместе с братом Виктории, а саму принцессу на старинном автомобиле привез ее отец, король Карл XVI Густав».* Данный фрагмент содержит информационно-статистические данные о ходе мероприятия без использования методов воздействия на искажение восприятия читателя.

Под пограничными случаями в рамках данного корпуса понимаются предложения, которые содержат отдельные лексико-дискурсивные маркеры, ассоциируемые с манипулятивными техниками, однако в конкретном контексте они не выполняют функции скрытого воздействия и потому были размечены как неманипулятивные.

Пример 5 (неманипулятивный фрагмент): *«Как пишет Bild, в результате аварии тяжелые травмы получили несколько человек».* Фраза содержит маркер источника, который формально может напоминать технику Appeal to Authority, а также устойчивое выражение «тяжелые травмы», по форме соответствующее эмоционально окрашенному эпитету (Loaded language). Вместе с тем данный фрагмент представляет собой нейтральное информационное сообщение с указанием источника и без выраженной оценочной нагрузки.

Пример 6 (неманипулятивный фрагмент): *«Лидер оппозиционной Либерально-демократической партии заявил, что не будет участвовать в голосовании».* Фраза содержит ссылку на политического актора и его заявление, а также предоставляет потенциально конфликтный контекст. В то же время само по себе высказывание не содержит интерпретации либо оценки действий политика, а также не является аргументом для поддержания авторской точки зрения.

Анализ пограничных случаев показывает, что наличие формальных лексических или синтаксических маркеров само по себе не является достаточным основанием для отнесения предложения к манипулятивным. Он также демонстрирует ключевой критерий экспертной разметки — функцию высказывания в контексте (используется ли соответствующая конструкция для изменения интерпретации или давления на мнение читателя). Однако, отдельные пограничные слу-

чаи могут допускать альтернативные трактовки, что в дальнейшем предполагается корректировать с помощью привлечения дополнительных валидирующих разметку экспертов.

Для оценки влияния синтетических данных на задачу бинарной классификации манипулятивных высказываний были проведены предварительные эксперименты с использованием линейных моделей. В качестве базового подхода использовалась линейная модель логистической регрессии с TF–IDF-представлением текста. Сравнивались два варианта обучения модели: обучение только на данных, размеченных вручную, и обучение на тех же данных с добавлением синтетических примеров, сгенерированных для бинарной классификации. Во всех экспериментах разбиение на обучающую и тестовую выборки выполнялось исключительно по оригинальным данным, синтетические примеры не включались в тестовую выборку, что исключало утечки информации. Качество классификации оценивалось с использованием метрики macro-F1.

Первоначальные эксперименты с одиночным разбиением на обучающую и тестовую выборки показали лишь незначительные различия между базовой моделью и моделью, обученной с использованием синтетических данных. В частности, значение macro-F1 для базовой модели составило 0.85, тогда как при добавлении синтетических примеров прирост оказался минимальным ($\Delta \approx +0.001$), что укладывается в пределы статистического шума. Однако применение стратифицированной кросс-валидации с пятью фолдами позволило выявить устойчивый положительный эффект от использования синтетических данных. Среднее значение macro-F1 для базовой модели составило 0.85 ± 0.04, тогда как модель, обученная на расширенной обучающей выборке, продемонстрировала значение 0.90 ± 0.04. Улучшение наблюдалось во всех фолдах кросс-валидации, что указывает на систематический характер эффекта. Полученные результаты свидетельствуют о том, что использование синтетических данных может способствовать повышению обобщающей способности моделей, по крайней мере бинарной классификации в условиях ограниченного объема данных. Отметим, что данные эксперименты носят предварительный характер и направлены прежде всего на оценку целесообразности применения синтетических данных в рамках дальнейших этапов исследования.

ЗАКЛЮЧЕНИЕ

Рассмотрены формирование специализированного корпуса предложений русскоязычных новостных текстов и разработка схемы разметки манипулятивных техник, ориентированная на задачу классификации по типам политических манипуляций на уровне предложений в русскоязычных новостных СМИ. В отличие от существующих корпусных ресурсов, разработанных преимущественно для англоязычных данных, предложенный подход учитывает языковую и контекстуальную специфику русскоязычных новостных медиа и адаптирован к особенностям реализации манипулятивных стратегий в данном контексте.

Предложенный корпус и схема разметки могут рассматриваться как эмпирическая основа для последующей разработки и тестирования методов автоматизированного выявления политических манипуляций в русскоязычных новостных СМИ. В дальнейшем развитие работы может быть связано с расширением корпуса, валидацией схемы разметки с участием нескольких экспертов, а также с применением предложенного ресурса в задачах автоматизированного анализа новостных текстов. Полученные результаты и разработанные инструменты могут быть использованы как в академических исследованиях, так и в прикладных задачах анализа медиаконтента.

СПИСОК ЛИТЕРАТУРЫ

1. *Entman R.M.* Framing: Toward clarification of a fractured paradigm // Journal of Communication. 1993. Vol. 43, No. 4, P. 51–58. https://doi.org/10.1111/j.1460-2466.1993.tb01304.x

2. *Chong D., Druckman J.N.* Framing theory // Annual Review of Political Science. 2007. Vol. 10. P. 103–126. https://doi.org/10.1146/annurev.polisci.10.072805.103054

3. *Mejias U.A., Vokuev N.E.* Disinformation and the media: The case of Russia and Ukraine // Media, Culture & Society. 2017. Vol. 39, No. 7. P. 1027–1042. https://doi.org/10.1177/0163443716686672

4. *Rozenas A., Stukal D.* How autocrats manipulate economic news: Evidence from Russia's state-controlled television // The Journal of Politics. 2019. Vol. 81, No. 3. P. 982–996. https://dx.doi.org/10.2139/ssrn.3023254

5. *Lazer D.M.J., Pentland A., Adamic L. et al*. Computational social science:

Obstacles and opportunities // Science. 2020. Vol. 369, No. 6507. P. 1060–1062. https://doi.org/10.1126/science.aaz8170

6. *Card D., Boydstun A.E., Gross J.H., Resnik P., Smith N.A.* The media frames corpus: Annotations of frames across issues // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. 2015. P. 438–444. https://doi.org/10.3115/v1/P15-2072

7. *Da San Martino G., Barrón-Cedeño A., Wachsmuth H., Nakov P.* Fine-grained analysis of propaganda in news articles // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019. P. 5636–5646. https://doi.org/10.18653/v1/D19-1565

8. *Field A., Atanasov P., Stukal D., Tucker J.A., Guess A.* Framing and agenda-setting in Russian news: A computational analysis of intricate political strategies // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2018. P. 3570–3580. https://doi.org/10.18653/v1/D18-1393

9. *Bhatia V., Chhaya N., Pala K., Bhargava P.* OpenFraming: Open-sourced tool for computational framing analysis of multilingual data // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2021. P. 242–250. https://doi.org/10.18653/v1/2021.emnlp-demo.28

10. *Card D., Paul M.J., Smith N.A.* Neural models for documents with metadata // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL). 2018. P. 2031–2040. https://doi.org/10.18653/v1/P18-1189

11. *Da San Martino G., Yu S., Barrón-Cedeño A., et. al.* SemEval-2020 Task 11: Detection of propaganda techniques in news articles // Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020). 2020. P. 1377–1414. https://doi.org/10.18653/v1/2020.semeval-1.186

12. *Kwak H., An J., Jing E.M., Ahn Y.* A systematic media frame analysis of 1.5 million New York Times articles from 2000 to 2017 // Proceedings of the 12th ACM Conference on Web Science. 2020. P. 305–314. https://doi.org/10.1145/3394231.3397921

13. *Kwak H., An J., Jing E.M., Ahn Y.* FrameAxis: Characterizing microframe bias and intensity with word embedding // PeerJ Computer Science. 2021. Vol. 7, Article e644. https://doi.org/10.7717/peerj-cs.644

14. *Entman R.M.* Framing bias: Media in the distribution of power // Journal of Communication. 2007. Vol. 57, No. 1. P. 163–173. https://doi.org/10.1111/j.1460-2466.2006.00336.x

CONSTRUCTION AND ANNOTATION OF A RUSSIAN-LANGUAGE NEWS CORPUS FOR AUTOMATED DETECTION OF POLITICAL MANIPULATION

N. L. Kulyulina[0009-0006-1715-1114]

Moscow Institute of Physics and Technology, Dolgoprudny, Russia

kuliulina.nl@phystech.edu

Abstract

This paper addresses the challenge of developing specialized corpus resources for the automated analysis of political manipulation in Russian-language media discourse. Although semantic text analysis and computational discourse analysis have advanced substantially in recent years, most existing corpora and annotation schemes are designed for English-language data and do not adequately capture the linguistic and discursive characteristics of Russian-language news media. The objective of this study is to construct a specialized corpus of Russian-language news texts and to develop an annotation scheme tailored to the automated analysis of political manipulation, with explicit consideration of the linguistic and discursive features of the Russian-language media environment. The study introduces a corpus of sentence-level fragments extracted from Russian-language news texts published between 2010 and 2019, together with an annotation scheme for manipulative techniques. The scheme is based on an adaptation of established international classifications of manipulative strategies and is reduced to a limited set of interpretable techniques relevant to Russian-language news discourse. The proposed framework covers emotional, argumentative, and contextual forms of manipulative influence. The resulting corpus and annotation scheme provide an empirical foundation for the development and evaluation of automated methods for analyzing political manipulation in Russian-language news media and may also support further research in media and political discourse.

Keywords: *media manipulation, Russian media, text corpus, data annotation, manipulative techniques, political communication, semantic analysis, computational discourse analysis.*

REFERENCES

1. *Entman R.M.* Framing: Toward clarification of a fractured paradigm // Journal of Communication. 1993. Vol. 43, No. 4. P. 51–58. https://doi.org/10.1111/j.1460-2466.1993.tb01304.x

2. *Chong D., Druckman J.N.* Framing theory // Annual Review of Political Science. 2007. Vol. 10. P. 103–126. https://doi.org/10.1146/annurev.polisci.10.072805.103054

3. *Mejias U.A., Vokuev N.E.* Disinformation and the media: The case of Russia and Ukraine // Media, Culture & Society. 2017. Vol. 39, No. 7. P. 1027–1042. https://doi.org/10.1177/0163443716686672

4. *Rozenas A., Stukal D.* How autocrats manipulate economic news: Evidence from Russia's state-controlled television // The Journal of Politics. 2019. Vol. 81, No. 3. P. 982–996. https://dx.doi.org/10.2139/ssrn.3023254

5. *Lazer D.M.J., Pentland A., Adamic L. et al*. Computational social science: Obstacles and opportunities // Science. 2020. Vol. 369, No. 6507. P. 1060–1062. https://doi.org/10.1126/science.aaz8170

6. *Card D., Boydstun A.E., Gross J.H., Resnik P., Smith N.A.* The media frames corpus: Annotations of frames across issues // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. 2015. P. 438–444. https://doi.org/10.3115/v1/P15-2072

7. *Da San Martino G., Barrón-Cedeño A., Wachsmuth H., Nakov P.* Fine-grained analysis of propaganda in news articles // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019. P. 5636–5646. https://doi.org/10.18653/v1/D19-1565

8. *Field A., Atanasov P., Stukal D., Tucker J.A., Guess A.* Framing and agenda-setting in Russian news: A computational analysis of intricate political strategies // Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2018. P. 3570–3580. https://doi.org/10.18653/v1/D18-1393

9. *Bhatia V., Chhaya N., Pala K., Bhargava P.* OpenFraming: Open-sourced tool for computational framing analysis of multilingual data // Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2021. P. 242–250. https://doi.org/10.18653/v1/2021.emnlp-demo.28

10. *Card D., Paul M.J., Smith N.A.* Neural models for documents with metadata // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL). 2018. P. 2031–2040. https://doi.org/10.18653/v1/P18-1189

11. *Da San Martino G., Yu S., Barrón-Cedeño A. et al.* SemEval-2020 Task 11: Detection of propaganda techniques in news articles // Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020). 2020. P. 1377–1414. https://doi.org/10.18653/v1/2020.semeval-1.186

12. *Kwak H., An J., Jing E.M., Ahn Y.* A systematic media frame analysis of 1.5 million New York Times articles from 2000 to 2017 // Proceedings of the 12th ACM Conference on Web Science. 2020. P. 305–314.
https://doi.org/10.1145/3394231.3397921

13. *Kwak H., An J., Jing E.M., Ahn Y.* FrameAxis: Characterizing microframe bias and intensity with word embedding // PeerJ Computer Science. 2021. Vol. 7, Article e644. https://doi.org/10.7717/peerj-cs.644

14. *Entman R.M.* Framing bias: Media in the distribution of power // Journal of Communication. 2007. Vol. 57, No. 1. P. 163–173.
https://doi.org/10.1111/j.1460-2466.2006.00336.x

СВЕДЕНИЯ ОБ АВТОРЕ

КУЛЮЛИНА Нина Леонидовна – студентка 2 курса магистерской программы Московского физико-технического института (национальный исследовательский университет) по направлению «Науки о данных»; научный сотрудник Центра изучения стабильности и рисков Национального исследовательского университета «Высшая школа экономики». Научные интересы включают применение передовых вычислительных методов анализа данных в сфере социальных наук, включая методы машинного обучения и большие языковые модели.

Nina Leonidovna KULYULINA – second-year student in the Master's program in Data Science at the Moscow Institute of Physics and Technology and a research fellow at the Center for Stability and Risk Analysis at the HSE University. Her research interests include the application of advanced computational methods of data analysis in social sciences, including machine learning methods and large-scale language models.

email: kuliulina.nl@phystech.edu

ORCID: 0009-0006-1715-1114

Материал поступил в редакцию 15 апреля 2026 года

УДК 004.8

ПРОЕКТИРОВАНИЕ И АВТОМАТИЧЕСКОЕ ПОСТРОЕНИЕ ГРАФА ЗНАНИЙ «МАТЕМАТИЧЕСКИЕ УРАВНЕНИЯ»

Е. К. Липачев[1] [0000-0001-7789-2332], Б. Р. Мурадымов[2] [0009-0004-1187-8158]

[1, 2]*Казанский (Приволжский) федеральный университет, г. Казань, Россия*

[1]*Университет Иннополис, г. Иннополис, Россия*

[1]elipachev@gmail.com, [2]muradymov.bulat@mail.ru

Аннотация

Предложен подход к проектированию и реализации графа знаний для представления и хранения знаний о математических уравнениях. В сформированном прототипе графа представлены знания об основных типах алгебраических уравнений, обыкновенных дифференциальных уравнений, уравнениях в частных производных и интегральных уравнениях. Граф знаний проектировался как математический артефакт экосистемы цифровой математической библиотеки Lobachevskii-DML, поэтому учитывались общие для экосистемы требования совместимости. Разработаны программные инструменты извлечения и обработки информации об уравнениях, представленной в цифровых библиотеках и электронных научных ресурсах. Прототип графа знаний сформирован на основе онтологии профессиональной математики OntoMath^{PRO} и таксономии уравнений, построенной на основе информации, извлеченной с веб-страниц научно-образовательного портала EqWorld «Мир математических уравнений». Онтология OntoMath^{PRO} расширена новыми классами уравнений и новыми отношениями для согласования с иерархией типов уравнений, представленной на портале EqWorld. Реализован комплекс программных модулей, обеспечивающих полный цикл формирования графа знаний: автоматическое извлечение сущностей из источников, связывание сущностей с концептами онтологии OntoMath^{PRO}, преобразование полученных знаний в RDF-представление с последующим сохранением в хранилище данных с возможностью выполнения SPARQL-запросов.

Ключевые слова: *граф знаний, извлечение знаний, математическое уравнение, математическая онтология, представление математического знания.*

ВВЕДЕНИЕ

Как известно, математические уравнения являются важнейшей составляющей математических документов. Алгебраические уравнения, обыкновенные дифференциальные уравнения, уравнения в частных производных, нелинейные уравнения, интегральные уравнения – это неполный список основных типов математических уравнений. Для каждого уравнения факты о нем в виде теорем, свойств, методов точного и приближенного решений представлены в многочисленных, не связанных между собой источниках, включая справочники, монографии и статьи. Значительная часть источников оцифрована и доступна в электронном варианте для чтения и обработки (см., например, [1–5]). Для формирования пространства математического знания как составляющей единого пространства научных знаний необходимо, в частности, решить задачи структурирования и семантического представления знаний (см., например, [6]). Создание математических онтологий направлено на решение этих задач (см., например, [7–9]). В работах [7, 10] проведены исследования по применению онтологий в прикладных задачах, представлены примеры использования математических онтологий в качестве компонент рекомендательных и поисковых систем.

Эффективным подходом к интеграции разнородных данных в единое пространство научных знаний являются графы знаний, в которых знания представлены в виде ориентированного графа сущностей и отношений между ними. Эффективность этого подхода подтверждается развитием глобального облака открытых связанных данных (Linked Open Data Cloud, https://lod-cloud.net/).

Термин «граф знаний» был введен в официальном блоге Google для обозначения использования в веб-поиске знаний, полученных с помощью связей между объектами [11]. Минимальный набор характеристик, позволяющий отличать графы знаний от других наборов знаний, описан в [12].

Для представления графов знаний используется стандарт консорциума W3C для моделирования и обмена данными Resource Description Framework (RDF, https://www.w3.org/RDF/), а также графы свойств (Property Graphs) (см., например, [13]).

Отметим ряд исследований, связанных с проектированием графов знаний в математической области. Подход к формированию графа знаний на основе онтологического описания научных предметных областей представлен в [14]. В качестве источника структурированных данных использована семантическая библиотека LibMeta (https://libmeta.ru/). Онтология содержимого семантической библиотеки применена в качестве средства формализации.

Технология построения графов знаний для необработанных научных текстов представлена в [15]. Предложена семантическая модель предметной области междисциплинарного научного журнала на основе онтологии библиотеки LibMeta. В этой работе показано, как можно перейти от неструктурированных текстов к тематическому анализу и встраиванию в граф знаний LibMeta.

В [16] предложен метод обогащения графа знаний и расширения предметной онтологии с помощью больших языковых моделей. В [17] представлены исследования процесса адаптации LLM к предметной области обыкновенных дифференциальных уравнений, представленной в виде онтологии русскоязычных ресурсов. Предложен интеграционный подход, рассматривающий ограничения больших языковых моделей в рамках библиотеки LibMeta, а также оценку релевантности ответа для различных LLM.

Настоящая работа является продолжением исследований, представленных в [18]. Предложены архитектура графа знаний «Математические уравнения» и алгоритм его реализации, основанный на извлечении знаний из научно-образовательного портала EqWorld «Мир математических уравнений» [2] и онтологии профессиональной математики OntoMath^{PRO} [7]. Использовались также сведения об уравнениях, представленных в справочниках по обыкновенным дифференциальным уравнениям и дифференциальным уравнениям в частных производных [19, 20], справочники по уравнениям математической физики [21, 22], справочник по интегральным уравнениям [23], а также переводы указанных книг на английский язык [24–28]. Для пополнения онтологии, которое также выполнено в работе, использовались сведения, представленные в Математической энциклопедии под редакцией И. М. Виноградова в пяти томах [29], а также перевод этой энциклопедии на английский язык [5, 30].

1. ПОСТАНОВКА ЗАДАЧИ

Настоящая работа посвящена разработке метода автоматического построения графа знаний «Математические уравнения» для представления и хранения знаний об основных типах математических уравнений.

Определение (см., например, [31]). *Граф знаний – это кортеж* $G = (E, R, T, D)$*, где E – множество вершин, представляющих сущности предметной области, R – множество отношений, связывающих сущности, T – множество RDF-триплетов* $(s, p, o) \in E \times R \times E$*, s – субъект, p – предикат, o – объект, D – множество описаний сущностей и отношений*.

Проектирование графа знаний «Математические уравнения» (далее «граф знаний») предполагает решение задач по нескольким взаимосвязанным направлениям:

- определение узлов графа знаний;
- определение отношений между узлами графа знаний;
- разработка программных инструментов для извлечения и импорта именованных сущностей и фактов из внешних источников;
- построение системы RML-правил отображения структурированных данных в RDF-формат;
- автоматическая генерация RDF-триплетов и формирование на их основе графа знаний;
- организация хранилища графа знаний с поддержкой SPARQL-запросов.

2. МЕТОД ФОРМИРОВАНИЯ ГРАФА ЗНАНИЙ

2.1. Архитектура системы построения графа знаний

На рис. 1 представлена архитектура системы автоматического построения графа знаний «Математические уравнения». Указаны основные источники знаний, а также основные модули, участвующие в формировании графа знаний.

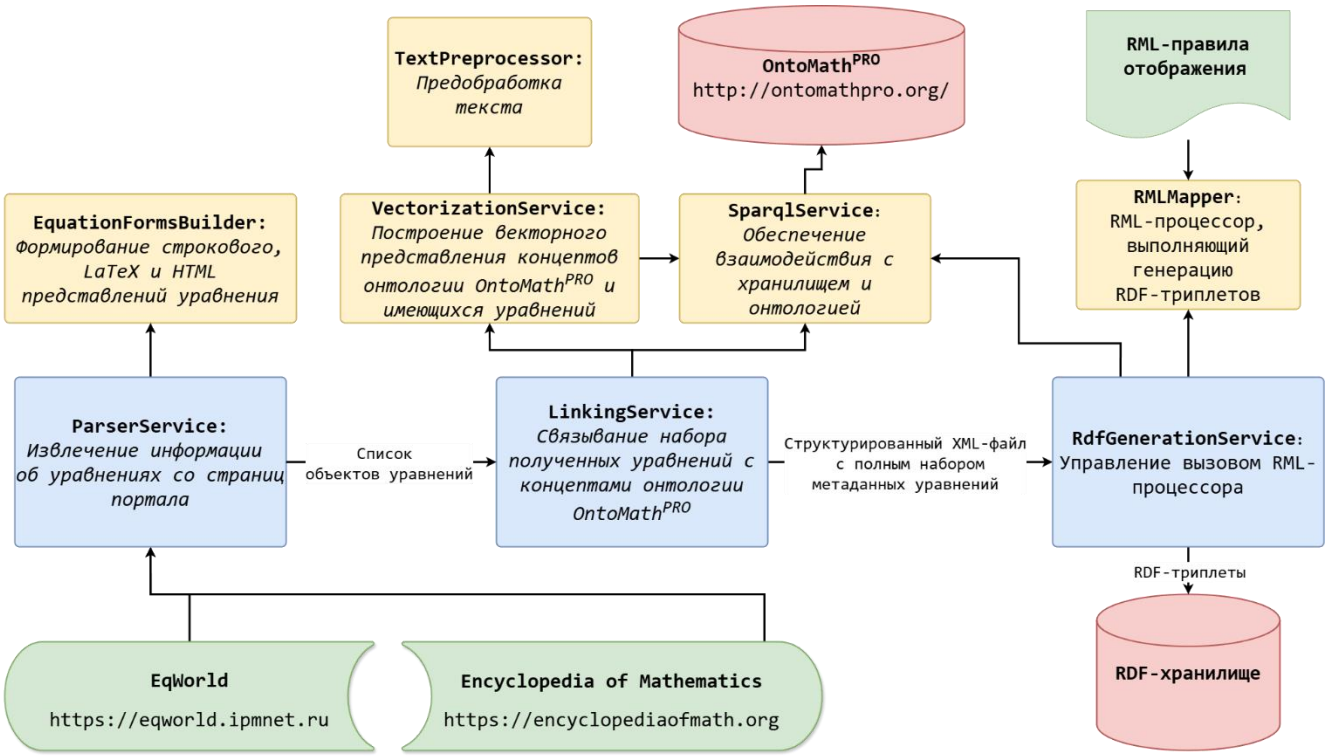


Рис. 1. Архитектура системы построения графа знаний «Математические уравнения».

2.2. Алгоритм извлечения знаний об уравнениях с портала EqWorld

Международный научно-образовательный портал «Мир математических уравнений» EqWorld содержит информацию о примерно 800 уравнениях, включая сведения о методах их решения и ссылки на внешние источники [2]. Представлена иерархия уравнений, распределенных по типам в 38 разделах и 60 подразделах.

Для автоматического сбора данных об уравнениях разработан алгоритм последовательного обхода 2000 веб-страниц портала, извлечения сведений об уравнениях и преобразования полученных данных в TEX-представление с последующим формированием объекта уравнения.

Алгоритм извлечения информации об уравнениях с портала EqWorld представлен в виде блок-схемы на рис. 2. Производится последовательный обход веб-страниц портала EqWorld с автоматическим поиском и извлечением данных о математических уравнениях, представленных на этих страницах. Далее орга-

низуется цикл, в котором обрабатывается информация о каждом уравнении. Извлекаются типы и подтипы уравнения, их названия, ссылки на PDF-документы. Производится преобразование извлеченной информации в $LaTeX$- и HTML- представления. В результате формируется список объектов уравнений.

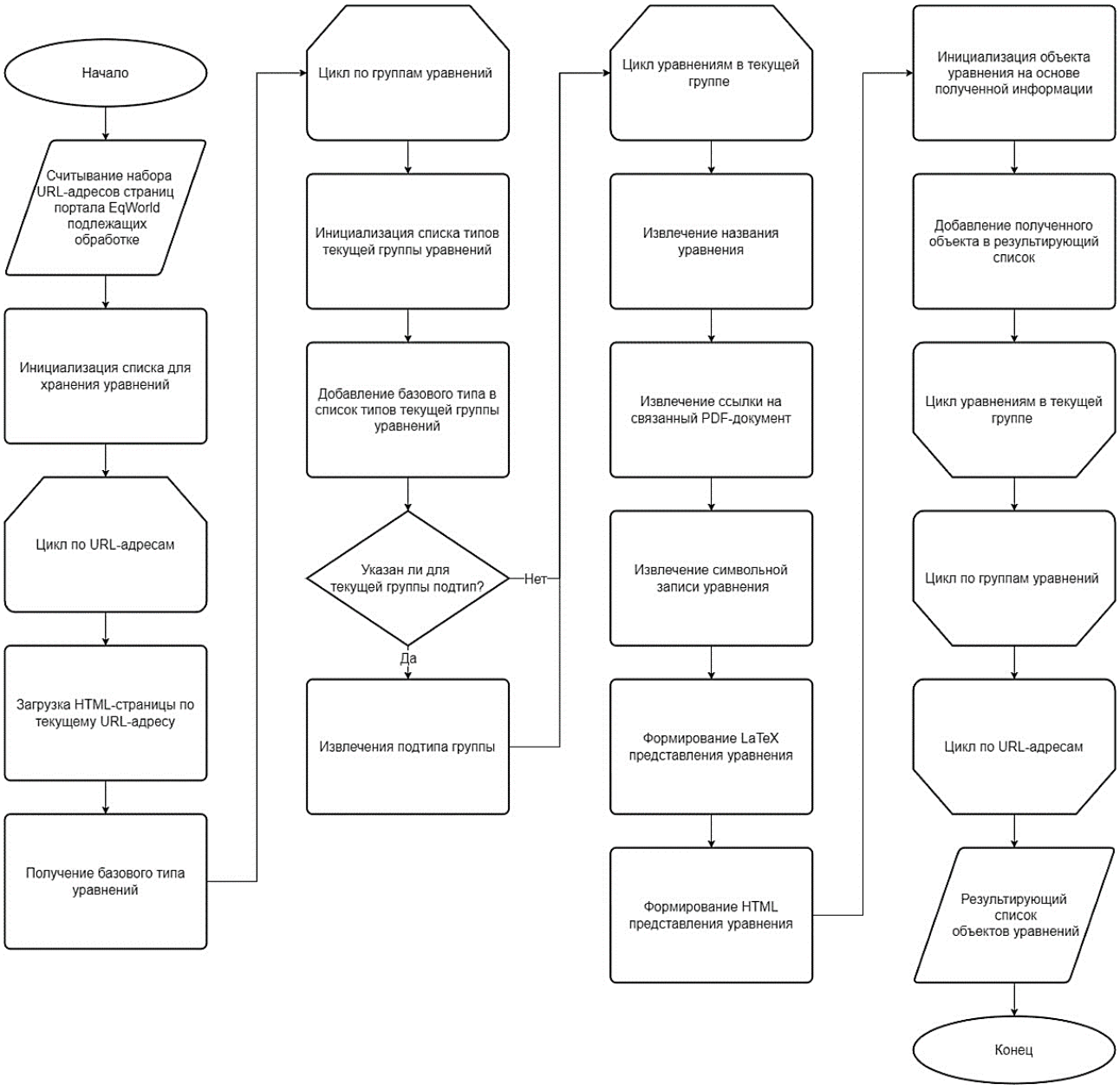


Рис. 2. Блок-схема алгоритма извлечения информации об уравнениях с веб-страниц портала EqWorld.

Далее представлен псевдокод алгоритма извлечения информации об уравнениях с веб-страниц портала EqWorld (алгоритм 1).

Алгоритм 1. Алгоритм извлечения информации об уравнениях.

```
    Входные данные: набор URL-адресов страниц, подлежащих обработке
    Выходные данные: Список уравнений
    # Чтение набора url-адресов из текстового файла
1   urls = GetUrls()
    # Инициализация результирующего списка
2   result = []
3   for url in urls:
      # Загрузка html-страницы по текущему url-адресу
4     htmlDoc = Load(url)
      # Извлечение базового типа уравнений (например, Обыкновенные диффе-
      ренциальные уравнения первого порядка)
5     baseType = GetBaseType(htmlDoc)
      # Извлечение групп уравнений
6     eqGroups = GetEquationsGroups(htmlDoc)
7     for eqGroup in eqGroups:
        # Формирование списка типов группы уравнений. Например, {Линейные
        обыкновенные дифференциальные уравнения второго порядка, Дифферен-
        циальные уравнения, содержащие степенные функции}
8       types = GetGroupTypes(baseType, eqGroup)
        # Получение списка уравнений группы
9       groupEquations = GetGroupEquations(eqGroup)
10      for equationContainer in groupEquations:
          # Извлечение ссылки на связанный pdf-документ
11        pdfLink = GetPdfLink(equationContainer)
          # Формирование строкового, LaTeX и html представлений уравнений
12        stringForm, latexForm, htmlForm = EquationFormsBuilder.GetEqua-
          tionForms(equationContainer)
          # Извлечение названий уравнения
13        label = GetLabel(equationContainer)
          # Создание объекта уравнения, на основе извлеченной информации
14        equationObject = new Equation(
            types,
            pdfLink,
            label,
```

```
            stringForm,
            latexForm,
            htmlForm)
          # Добавление объекта уравнений в результирующий список
          result.Add(equationObject)
      end for
    end for
  end for
  return result
```

2.3. Алгоритм извлечения знаний из онтологии OntoMath^{PRO}

Онтология математического знания OntoMath^{PRO} разработана с целью классификации и систематизации основных понятий профессиональной математики [7, 32]. В этой онтологии представлены основные разделы математики, включая, в частности, таксономии «Уравнение», «Элемент теории дифференциальных уравнений», «Уравнение математического анализа», «Уравнение численного анализа», используемые в качестве источников фактов об уравнениях в процессе формирования графа знаний [33].

Для соответствия типов уравнений, представленных на портале EqWorld, онтология OntoMath^{PRO} была пополнена новыми классами уравнений. Созданы новые отношения в онтологии: `omp2:ParticularSolution`, `omp2:ExactSolution`. Эти связи позволили связать каждое уравнение с методами его решения и точным решением.

Разработан алгоритм связывания информации о классах уравнений, представленной в онтологии OntoMath^{PRO}, с информацией об уравнениях, извлеченной с портала EqWorld. Под связыванием понимается поиск в онтологии концепта, подходящего для конкретного уравнения, присваивание объекту уравнения URI найденного концепта и присоединение к объекту уравнения сведений, полученных из онтологии.

Алгоритм состоит из двух этапов. На первом этапе производится поиск подходящего концепта онтологии на основе точного соответствия названия или названий типов связываемого уравнения (см. Алгоритм 2).

Алгоритм 2. Метод прямого поиска.

```
    Входные данные: Объект уравнения
    Выходные данные: Логическое значение (true\false)
    # Если у уравнения есть название, но изначально осуществляем поиск
    по нему
1   if equation.Label != 'Без названия':
      # Запрос к онтологии на поиск концепта по переданному названию
2     result = SparqlService.GetConceptByLabel(equation.Label)
      # Если концепт найден
3     if result != null:
        # Дополнение объекта уравнения информацией, имеющейся в онто-
        логии
4       CompleteEquation(equation, result)
5       return true
    # Если поиск по названию не удался, то последовательно выполняется
    поиск по названиям типов уравнения
6   for eqType in equation.Types:
7     result = SparqlService.GetConceptByLabel(eqType)
8     if result != null:
9       CompleteEquation(equation, result)
10      return true
    # Поиск не дал результатов
11  return false
```

Если поиск не дает результата, например, из-за различий в названии типа уравнения, на втором этапе выполняется процедура семантического сопоставления названий. Концепты онтологии предварительно векторизуются по схеме TF-IDF. Затем из названия и типов уравнения формируется текстовый документ, который проходит аналогичную векторизацию, после это наиболее близкий концепт определяется по косинусной мере сходства.

Блок-схема алгоритма представлена на рис. 3. По этому алгоритму производится последовательная обработка входного списка уравнений с целью сопоставления каждого уравнения с соответствующим концептом онтологии. Для каждого уравнения сначала предпринимается попытка прямого поиска кон-

цепта по названию уравнения, а затем выполняется поиск по его типам и подтипам, перебираемым от частного к общему. В случае успешного нахождения концепта уравнению присваиваются его URI и дополнительные сведения из онтологии. Если же подходящий концепт не был найден, применяется метод семантического сопоставления: из названия и типов уравнения формируется текстовый документ, к которому применяются предобработка и векторизация на основе схемы TF-IDF, далее вычисляется значение меры сходства и определяется наиболее близкий концепт онтологии. В результате формируется XML-файл, содержащий список уравнений.

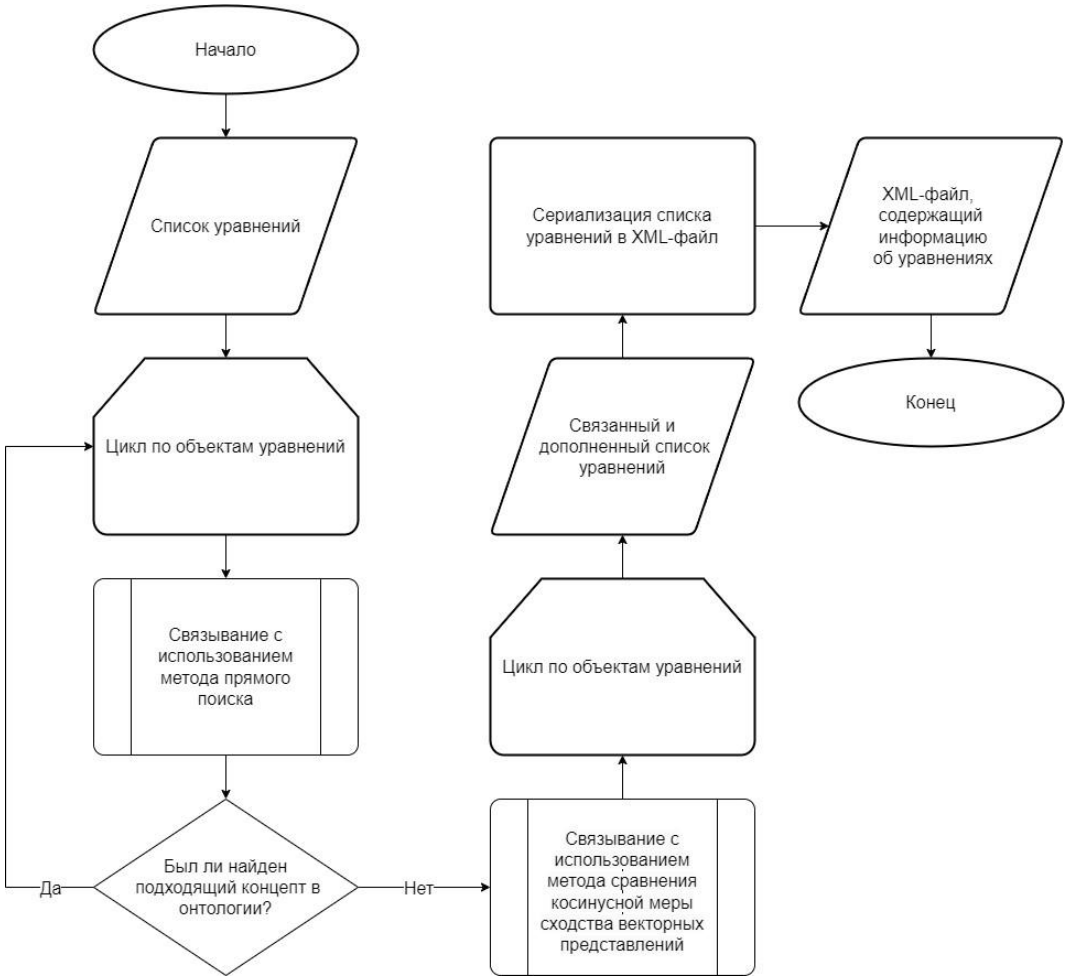


Рис. 3. Блок-схема алгоритма связывания объектов уравнений с концептами онтологии OntoMath^{PRO}.

Ниже представлен псевдокод алгоритма связывания объектов уравнений с концептами онтологии (Алгоритм 3).

Алгоритм 3. Метод связывания объектов уравнений с концептами онтологии

Входные данные: *Список объектов уравнений*
Выходные данные: *XML-файл, содержащий информацию об уравнениях, размеченных в терминах концептов онтологии*

```
for equation in equations:
    # Изначально происходит попытка прямого поиска
    if TryDirectLinking(equation):
        continue
    # Получение наиболее близкого концепта, на основе оценки косинусной меры сходства векторных представлений концептов онтологии и объекта уравнения
    closestConcept = Vectorizer.GetClosestConcept(equation)
    # Извлечение информации по найденному концепту
    conceptInfo = SparqlService.GetConceptInfo(closestConcept)
    # Дополнение объекта уравнения найденной информацией
    CompleteEquation(equation, conceptInfo)
# Сохранение результирующего списка уравнений в виде XML-файла
SaveAsXML(equations)
```

2.4. Алгоритм генерации RDF-триплетов

В результате выполнения операций, представленных в п. 2.2 и 2.3, формируется XML-файл, содержащий метаданные уравнений. Для каждого объекта уравнений в XML-файле, в частности, зафиксированы следующие метаданные:

- Названия уравнения на русском и английском языках (xml-тэги `LabelText` и `LangTag`);
- Тип уравнения (xml-тэг `Types`);
- Строковое, HTML и $LaTeX$ представления формульной записи уравнения (тэги `StringFrom`, `HtmlForm`, `LatexForm`);
- URI родительского и рассматриваемого концептов онтологии (`SubClassOf`, `LinkedOmpConcept`);
- Описание свойств уравнения (тэг `Comments`).

Фрагмент XML-файла приведен на рис. 4.

Преобразование XML-данных в RDF-представление выполняется на основании разработанной системы правил, записанной на языке RML (RDF Mapping

Language) [34]. Система RML-правил задает карты троек, однозначно определяющие субъект, предикат и объект каждого генерируемого триплета (фрагмент карты представлен на рис. 5).

Набор правил отображения и XML-файл уравнений передаются RML-процессору, который генерирует триплеты и записывает их файл в формате Turtle (фрагмент файла представлен на рис. 6).

```
<Equation>
  <URI>9ef5f5b0-e176-462d-9d70-5a54d54bc895</URI>
  <Labels>
    <Label>
      <LabelText>Уравнение Абеля второго рода</LabelText>
      <LangTag>ru</LangTag>
    </Label>
    <Label>
      <LabelText>Abel's differential equation of the second kind</LabelText>
      <LangTag>en</LangTag>
    </Label>
    <Label>
      <LabelText>Дифференциальное уравнение Абеля 2-го рода</LabelText>
      <LangTag>ru</LangTag>
    </Label>
    <Label>
      <LabelText>Абеля дифференциальное уравнение 2-го рода</LabelText>
      <LangTag>ru</LangTag>
    </Label>
  </Labels>
  <Types>
    <string>Обыкновенные дифференциальные уравнения первого порядка</string>
  </Types>
  <SubClassOf>http://ontomathpro.org/omp2#AbelDifferentialEquation</SubClassOf>
  <LinkedOmpConcept>http://ontomathpro.org/omp2#Abel%27sDifferentialEquationOfTheSecondKind<
  <StringForm>yy'=f(x)y^(2)+g(x)y+h(x).</StringForm>
  <Reference>https://eqworld.ipmnet.ru/en/solutions/ode/ode0126.pdf</Reference>
  <Comments>
    <string>Abel's differential equation of the first kind is an equation of the form
$$
y' = f_0(x) + f_1(x)y + f_2(x)y^2 + f_3(x)y^3.
$$

Abel's differential equations of the first kind represent a natural generalization of the Ricc
```

Рис. 4. Фрагмент XML-файла с информацией об уравнении Абеля 2-го рода.

```
<#EquationMapping> a rr:TriplesMap;
  rml:logicalSource [
    rml:source "equations.xml" ;
    rml:iterator "/ArrayOfEquation/Equation";
    rml:referenceFormulation ql:XPath;
  ];

  rr:subjectMap [
    rr:template "http://www.eqgraph.ru/{URI}";
  ];

  rr:predicateObjectMap [
  rr:predicate rdf:type;
  rr:objectMap [
    rml:reference "LinkedOmpConcept";
    rr:termType rr:IRI;
    ]
  ];
```

Рис. 5. Фрагмент карты троек, в которой определены логический источник, карта субъекта и карта предиката-объекта.

```
<http://www.eqgraph.ru/9ef5f5b0-e176-462d-9d70-5a54d54bc895> a <http://ontomathpro.org/omp2#Abel%27sDifferentialEquationOfTheSecondKind>;
  eq:htmlForm "<i>yy</i><span>&prime;</span> =<i>f</i><span>(<i>x</i>)</span><i>y</i><sup>2</sup> +<i>g</i>(<i>x</i>)<span><i>y</i></span
  eq:latexForm "\\[y\\frac{dy}{dx}=f(x)y^{2}+g(x)y+h(x)\\]";
  eq:reference "https://eqworld.ipmnet.ru/en/solutions/ode/ode0126.pdf";
  eq:stringForm "yy'=f(x)y^(2)+g(x)y+h(x).";
  rdfs:comment """Abel's differential equation of the second kind is an equation of the form
$$
\left(g_0(x) + g_1(x)y \right)y' = f_0(x) + f_1(x)y + f_2(x)y^2 + f_3(x)y^3.
$$

If $g_0,g_1 \in C^1(a,b)$ and $g_1(x) \neq 0$, $g_0(x) + g_1(x)y \neq 0$, Abel's differential equation of the second kind can be reduced

Abel's differential equations of the first and second kinds, as well as their further generalizations""";
  rdfs:label "Abel's differential equation of the second kind"@ru, "Абеля дифференциальное уравнение 2-го рода"@ru,
    "Дифференциальное уравнение Абеля 2-го рода"@ru, "Уравнение Абеля второго рода"@ru;
```

Рис. 6. Фрагмент файла, содержащего сгенерированные RDF-триплеты для сущности «Уравнение Абеля второго рода».

Блок-схема алгоритма генерации RDF-триплетов представлена на рис. 7. Производится преобразование структурированных данных об уравнениях в формат RDF с последующей записью в граф знаний. Для этого формируются RML-правила отображения, на основе которых настраивается RML-процессор, получающий на вход XML-файл с описанием уравнений. После вызова процессора сгенерированные RDF-триплеты сохраняются в хранилище Open Link Virtuoso в виде графа знаний. Созданное хранилище предоставляет возможность выполнения SPARQL-запросов.

На Рис. 8 приведен фрагмент графа знаний, представляющий узлы и отношения для сущности «Уравнение Абеля второго рода.

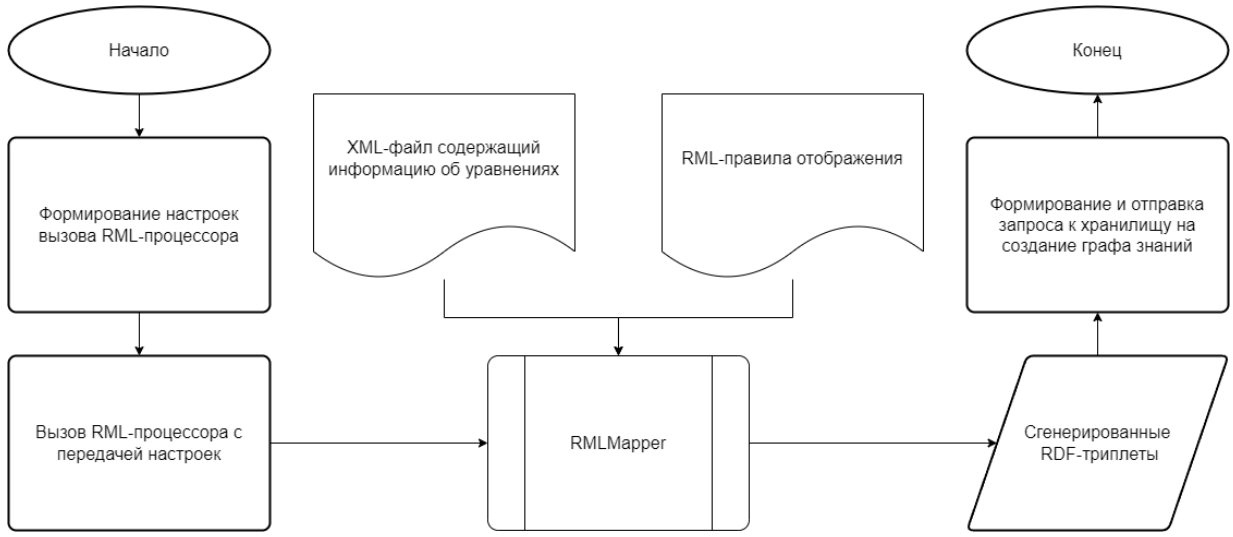


Рис. 7. Блок-схема алгоритма генерации RDF-триплетов.

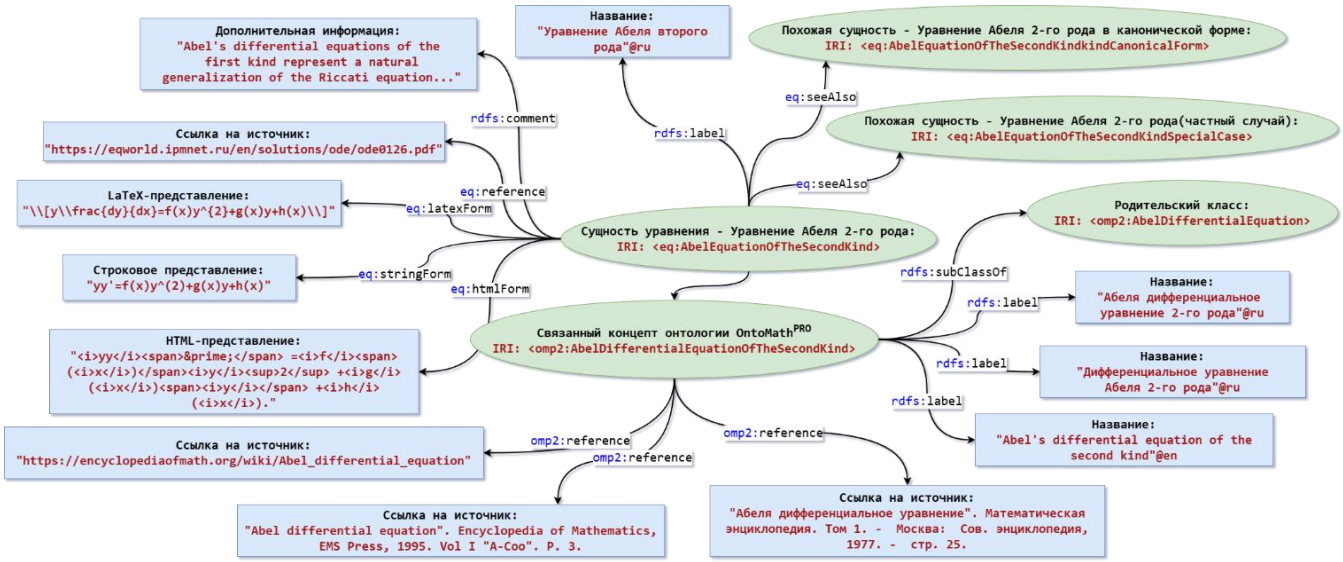


Рис. 8. Визуализация фрагмента графа знаний, представляющего узлы и отношения для сущности «Уравнение Абеля второго рода».

3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ

Разработанный программный комплекс написан на языке C# с применением платформы .NET и состоит из трех основных модулей и набора вспомогательных программных сервисов.

Модуль `ParserService` содержит функционал, обеспечивающий последовательный обход веб-страниц портала EqWorld и извлечение информации об

уравнениях. Для извлечения данных с веб-страниц (веб-скрейпинг) используются функции библиотеки HTMLAgilityPack (https://html-agility-pack.net/). Каждое обнаруженное уравнение представляется экземпляром класса `Equation`.

Модуль `LinkingService` обеспечивает дополнение данных с помощью сопоставления уравнений с концептами онтологии OntoMath^{PRO}. Для векторизации и вычисления семантической близости задействована библиотека машинного обучения Accord.Net (https://accord-framework.net/). Взаимодействие с онтологией осуществляется через систему сформированных SPARQL-запросов.

Модуль `RdfGenerationService` координирует работу RML-процессора: передает ему XML-файл с описаниями уравнений и набор правил отображения, после чего сохраняет сгенерированное множество RDF-триплетов в хранилище графа знаний.

Вспомогательный уровень состоит из следующих программных сервисов:

- `SparqlService` – формирование, отправка запросов к онтологии и обработка полученных ответов;
- `VectorizationService` – построение TF-IDF-векторных представлений концептов онтологии;
- `TextPreprocessor` – предобработка текстовых данных (токенизация, лемматизация и пр.);
- `EquationFormsBuilder` – формирование строкового, HTML- и $LaTeX$-представлений уравнений.

ЗАКЛЮЧЕНИЕ

Предложен метод и реализован алгоритм построения графа знаний «Математические уравнения», объединяющий факты об уравнениях, представленные на научно-образовательном портале EqWorld «Мир математических уравнений» и онтологии профессиональной математики OntoMath^{PRO}. Сформированный граф включен в цифровую экосистему OntoMath [35] цифровой математической библиотеки Lobachevskii-DML [36, 37] в качестве математического артефакта, что определяет его практическую значимость.

Дальнейшее развитие работы предполагает значительное пополнение графа знаний, включая добавление новых классов уравнений, введение допол-

нительных типов отношений между ними, а также установление связей с другими объектами знаний математического пространства.

Благодарности

Выражаем благодарность Наталии Павловне Тучковой, Александру Михайловичу Елизарову и Ольге Авенировне Невзоровой за проявленный интерес к исследованию, значимые замечания и советы при оформлении статьи.

СПИСОК ЛИТЕРАТУРЫ

1. Общероссийский портал Math-Net.Ru. URL: https://www.mathnet.ru/ (дата обращения 22.03.2026).

2. EqWorld. «Мир математических уравнений».
URL: https://eqworld.ipmnet.ru/ (дата обращения 22.03.2026).

3. Цифровая математическая библиотека Lobachevskii-DML.
URL: https://lobachevskii-dml.ru/ (дата обращения 22.03.2026).

4. Numdam, the French digital mathematics library.
URL: https://www.numdam.org/ (дата обращения 22.03.2026).

5. Encyclopedia of Mathematics. URL: https://encyclopediaofmath.org/ (дата обращения 22.03.2026).

6. *Elizarov A.M., Lipachev E.K.* Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge // Scientific and Technical Information Processing. 2023. Vol. 50, No. 1. P. 35–39. https://doi.org/10.3103/s0147688223010021

7. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A.* OntoMathPRO: an Ontology of Mathematical Knowledge // Doklady Mathematics. 2022. Vol. 106, No. 3. P. 429–435. https://doi.org/10.1134/S1064562422700016

8. *Муромский А.А., Тучкова Н.П.* Представление математических понятий в онтологии научных знаний // Онтология проектирования. 2019. Т. 9, №1 (31). С. 50–69. https://doi.org/10.18287/2223-9537-2019-9-1-50-69

9. *Nevzorova O.A., Falileeva M.V., Kirillovich A.V. et al.* OntoMathEdu Educational Ontology: Problems of Ontological Engineering // Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications. 2023. Vol. 33, No. 3. P. 460–466. https://doi.org/10.1134/S1054661823030367

10. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* Ontological Approach:

Knowledge Representation and Knowledge Extraction // Lobachevskii J. Math. 2020. Vol. 41 (10). P. 1938–1948. https://doi.org/10.1134/S1995080220100030

11. *Singhal A.* Introducing the Knowledge Graph: things, not strings // Google Official Blog, 2012.
URL: https://blog.google/products/search/introducing-knowledge-graph-things-not/ (дата обращения 22.03.2026).

12. *Paulheim H.* Knowledge graph refinement: A survey of approaches and evaluation methods // Semantic Web. 2017. Vol. 8. P. 489–508.
https://doi.org/10.3233/SW-160218

13. *Hogan A., Gutierrez C., Cochez M. et al.* Data Graphs // In: Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge. Springer, Cham, 2022. P. 5–23. https://doi.org/10.1007/978-3-031-01918-0_2

14. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* Ontological Approach to a Knowledge Graph Construction in a Semantic Library // Lobachevskii J. Math. 2023. Vol. 44, No. 6. P. 2229–2239. https://doi.org/10.1134/S1995080223060471

15. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* From Texts to Knowledge Graph in the Semantic Library LibMeta // Lobachevskii J. Math. 2024. Vol. 45, No. 5. P. 2211–2219. https://doi.org/10.1134/S1995080224602625

16. *Халов А.П., Атаева О.М.* Автоматические и полуавтоматические методы построения графа знаний предметной области и расширения онтологии // Электронные библиотеки. 2025. Т. 28. №. 6. С. 1481–1519.
https://doi.org /10.26907/1562-5419-2025-28-6-1481-1519

17. *Ataeva O.M., Tuchkova N.P.* Navigation with Large Language Models in Subject Domain of Ordinary Differential Equation // Lobachevskii J. Math. 2025. Vol. 46, No. 6. P. 2723–2735. https://doi.org/10.1134/S1995080225608227

18. *Липачев Е.К., Мурадымов Б.Р.* На пути к построению графа знаний математических уравнений // Системы высокой доступности. 2026. Т. 22, № 1. С. 41–46. https://doi.org/10.18127/j20729472-202601-08

19. *Зайцев В.Ф., Полянин А.Д.* Справочник по обыкновенным дифференциальным уравнениям. М.: Физматлит, 2001. 576 с.

20. *Зайцев В.Ф., Полянин А.Д.* Справочник по дифференциальным уравнениям с частными производными первого порядка. М.: Физматлит, 2003. 416 с.

21. *Полянин А.Д.* Справочник по линейным уравнениям математической физики. М.: Физматлит, 2001. 575 с.

22. *Полянин А.Д., Зайцев В.Ф.* Нелинейные уравнения математической физики. М.: Юрайт, 2017. 432 с.

23. *Полянин А.Д., Манжиров А.В.* Справочник по интегральным уравнениям. М.: Физматлит, 2003. 369 с.

24. *Polyanin A.D., Zaitsev V.F.* Handbook of Ordinary Differential Equations: Exact Solutions, Methods, and Problems. CRC Press/Chapman and Hall, 2017. 1496 p. https://doi.org/10.1201/9781315117638

25. *Polyanin A.D., Zaitsev V.F.* Handbook of Exact Solutions for Ordinary Differential Equations, 2nd Edition (Updated and Extended). CRC Press, Boca Raton–New York, 2003. 816 p.

26. *Polyanin A.D., Zaitsev V.F., Moussiaux A.* Handbook of First Order Partial Differential Equations. CRC Press, 2001. 520 p. https://doi.org/10.1201/b16828

27. *Polyanin A.D., Zaitsev V.F.* Handbook of Nonlinear Partial Differential Equations, Second edition. CRC Press, 2012. 1912 p. https://doi.org/10.1201/b11412

28. *Polyanin A.D., Manzhirov A.V.* Handbook of Integral Equations, 2nd Edition. Chapman and Hall/CRC Press, Boca Raton–London, 2008. 1144 p. https://doi.org/10.1201/9781420010558

29. *Виноградов И.М. (Ред.)* Математическая энциклопедия (в 5 томах) М.: Советская энциклопедия (1977–1985).

30. *Hazewinkel M.* (Ed.) Encyclopaedia of Mathematics. An updated and annotated translation of the Soviet ‘Mathematical Encyclopaedia’. Vol. 1–10. Springer Dordrecht, 1988. https://doi.org/10.1007/978-94-009-6000-8

31. *Ji S., Pan S., Cambria E., Marttinen P., Yu P.S.* A Survey on Knowledge Graphs: Representation, Acquisition, and Applications // IEEE Transactions on Neural Networks and Learning Systems. 2022. Vol. 33, No. 2. P. 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

32. *Kirillovich A.V., Nevzorova O.A., Lipachev E.K.* OntoMathPRO 2.0 Ontology: Up-dates of Formal Model // Lobachevskii J. of Math. 2022. Vol. 43, No. 12. P. 3504–3514. https://doi.org/10.1134/S1995080222150136

33. *Елизаров А.М., Кириллович А.В., Липачев Е.К., Невзорова О.А.* Новые

компоненты онтологии OntoMathPRO представления математического знания // Научный сервис в сети Интернет. 2023. № 25. С. 141–151. https://doi.org/10.20948/abrau-2023-32

34. RDF Mapping Language (RML). Unofficial Draft, 20 June 2024. URL: https://rml.io/specs/rml/ (дата обращения 22.03.2026)

35. *Елизаров А.М., Кириллович А.В., Липачев Е.К., Невзорова О.А.* Цифровая экосистема OntoMath как подход к построению пространства математических знаний // Электронные библиотеки. 2023. Т. 26. № 2. С. 154–202. https://doi.org/10.26907/1562-5419-2023-26-2-154-202

36. *Elizarov A.M., Lipachev E.K.* Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge // Scientific and Technical Information Processing. 2023. Vol. 50, No. 1. P. 35–39. https://doi.org/10.3103/s0147688223010021

37. *Elizarov A., Lipachev E.* Big math methods in Lobachevskii-DML // CEUR Workshop Proc. 2019. Vol. 2523. P. 59–72. https://ceur-ws.org/Vol-2523/invited08.pdf, last accessed 2026/04/04

ENGINEERING AND AUTOMATIC CONSTRUCTION OF A KNOWLEDGE GRAPH “MATHEMATICAL EQUATIONS”

E. K. Lipachev[1] [0000-0001-7789-2332], B. R. Muradymov[2] [0009-0004-1187-8158]

[1, 2] Kazan (Volga region) Federal University, Kazan, Russia

[1]Innopolis University, Innopolis, Russia

[1]elipachev@gmail.com, [2]muradymov.bulat@mail.ru

Abstract

We propose an approach to engineering and implementing a knowledge graph for representing and storing knowledge about mathematical equations. We have developed a knowledge graph prototype that represents knowledge about the main types of mathematical equations, including algebraic equations, ordinary differential equations, partial differential equations, and integral equations. We designed the knowledge graph of mathematical equations as a mathematical artifact. We are inte-

grating this artifact into the digital ecosystem of the Lobachevskii Digital Mathematical Library, therefore, we took into account the ecosystem's general compatibility requirements during the design. We have developed software tools for extracting and processing information about equations presented in digital libraries and electronic scientific resources. The current version of the knowledge graph prototype is based on the OntoMathPRO ontology of professional mathematics and a taxonomy of equations, built on information extracted from the web pages of the portal EqWorld "The World of Mathematical Equations." We expanded the OntoMathPRO ontology with new equation classes and new relationships to align with the equation type hierarchy presented on the EqWorld portal. We implemented a set of software modules that support the full cycle of knowledge graph generation, including a module for automatically extracting entities from external sources, a module for linking entities to OntoMathPRO ontology concepts, and a module for converting the acquired knowledge into an RDF representation and then storing it in a data warehouse. The knowledge graph supports SPARQL queries.

Keywords: *knowledge graph, knowledge extraction, mathematical equation, mathematical ontology, representation of mathematical knowledge.*

REFERENCES

1. All-Russian Portal Math-Net.Ru. URL: https://www.mathnet.ru/ (Accessed: 22.03.2026).

2. EqWorld. The World of Mathematical Equations. URL: https://eqworld.ipmnet.ru/ (Accessed: 22.03.2026).

3. Lobachevskii Digital Mathematical Library. URL: https://lobachevskii-dml.ru/ (Accessed: 22.03.2026).

4. Numdam, the French digital mathematics library. URL: https://www.numdam.org/ (Accessed: 22.03.2026).

5. Encyclopedia of Mathematics. URL: https://encyclopediaofmath.org/ (Accessed: 22.03.2026).

6. *Elizarov A.M., Lipachev E.K.* Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge // Scientific and Technical Information Processing. 2023. Vol. 50, No. 1. P. 35–39. https://doi.org/10.3103/s0147688223010021

7. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A.* OntoMathPRO:

an Ontology of Mathematical Knowledge // Doklady Mathematics. 2022. Vol. 106, No. 3. P. 429–435. https://doi.org/10.1134/S1064562422700016

8. *Muromskiy A.A., Tuchkova N.P.* Representation of Mathematical Concepts in the Ontology of Scientific Knowledge // Ontology of Designing. 2019. Vol. 9, No. 1 (31). P. 50–69. https://doi.org/10.18287/2223-9537-2019-9-1-50-69

9. *Nevzorova O.A., Falileeva M.V., Kirillovich A.V.et al.* OntoMathEdu Educational Ontology: Problems of Ontological Engineering // Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications. 2023. Vol. 33, No. 3. P. 460–466. https://doi.org/10.1134/S1054661823030367

10. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* Ontological Approach: Knowledge Representation and Knowledge Extraction // Lobachevskii J. Math. 2020. Vol. 41 (10). P. 1938–1948. https://doi.org/10.1134/S1995080220100030

11. *Singhal A.* Introducing the Knowledge Graph: things, not strings // Google Official Blog, 2012. URL: https://blog.google/products/search/introducing-knowledge-graph-things-not/ (Accessed: 22.03.2026)

12. *Paulheim H.* Knowledge graph refinement: A survey of approaches and evaluation methods // Semantic Web. 2017. Vol. 8. P. 489–508. https://doi.org/10.3233/SW-160218

13. *Hogan A., Gutierrez C., Cochez M. et al.* Data Graphs // In: Knowledge Graphs. Synthesis Lectures on Data, Semantics, and Knowledge. Springer, Cham, 2022. P. 5–23. https://doi.org/10.1007/978-3-031-01918-0_2

14. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* Ontological Approach to a Knowledge Graph Construction in a Semantic Library // Lobachevskii J. Math. 2023. Vol. 44, No. 6. P. 2229–2239. https://doi.org/10.1134/S1995080223060471

15. *Ataeva O.M., Serebryakov V.A., Tuchkova N.P.* From Texts to Knowledge Graph in the Semantic Library LibMeta // Lobachevskii J. Math. 2024. Vol. 45, No. 5. P. 2211–2219. https://doi.org/10.1134/S1995080224602625

16. *Khalov A.P., Ataeva O.M.* Automatic and Semi-automatic Methods for Domain Knowledge-Graph Construction and Ontology Expansion // Russian Digital Libraries Journal. 2025. Vol. 28, No. 6. P. 1481–1519. https://doi.org /10.26907/1562-5419-2025-28-6-1481-1519

17. *Ataeva O.M., Tuchkova N.P.* Navigation with Large Language Models in Subject Domain of Ordinary Differential Equation // Lobachevskii J. Math. 2025. Vol. 46, No. 6. P. 2723–2735. https://doi.org/10.1134/S1995080225608227

18. *Lipachev E.K., Muradymov B.R.* Towards Building the Knowledge Graph of Mathematical Equations // Highly Available Systems. 2026. Vol. 22, No. 1. P. 41−46. https://doi.org/10.18127/j20729472-202601-08

19. *Zajcev V.F., Polyanin A.D.* Spravochnik po obyknovennym differenci-al'nym uravneniyam. M.: Fizmatlit, 2001. 576 s.

20. *Zajcev V.F., Polyanin A.D.* Spravochnik po differencial'nym uravneni-yam s chastnymi proizvodnymi pervogo poryadka. M.: Fizmatlit, 2003. 416 s.

21. *Polyanin A.D.* Spravochnik po linejnym uravneniyam matematicheskoj fiziki. M.: Fizmatlit, 2001. 575 s.

22. *Polyanin A.D., Zajcev V.F.* Nelinejnye uravneniya matematicheskoj fiziki. M.: Yurajt, 2017. 432 s.

23. *Polyanin A.D., Manzhirov A.V.* Spravochnik po integral'nym uravneniyam. M.: Fizmatlit, 2003. 369 s.

24. *Polyanin A.D., Zaitsev V.F.* Handbook of Ordinary Differential Equations: Exact Solutions, Methods, and Problems. CRC Press/Chapman and Hall, 2017. 1496 p. https://doi.org/10.1201/9781315117638

25. *Polyanin A.D., Zaitsev V.F.* Handbook of Exact Solutions for Ordinary Differential Equations, 2nd Edition (Updated and Extended). CRC Press, Boca Raton–New York, 2003. 816 p.

26. *Polyanin A.D., Zaitsev V.F., Moussiaux A.* Handbook of First Order Partial Differential Equations. CRC Press, 2001. 520 p. https://doi.org/10.1201/b16828

27. *Polyanin A.D., Zaitsev V.F.* Handbook of Nonlinear Partial Differential Equations, Second edition. CRC Press, 2012. 1912 p. https://doi.org/10.1201/b11412

28. *Polyanin A.D., Manzhirov A.V.* Handbook of Integral Equations, 2nd Edition. Chapman & Hall/CRC Press, Boca Raton–London, 2008. 1144 p. https://doi.org/10.1201/9781420010558

29. *Vinogradov I.M.* (Red.) Matematicheskaya ehnciklopediya (v 5 tomah) M.: Sovetskaya ehnciklopediya (1977–1985).

30. *Hazewinkel M. (Ed.)* Encyclopaedia of Mathematics. An updated and annotated translation of the Soviet 'Mathematical Encyclopaedia'. Vol. 1–10. Springer Dordrecht, 1988. https://doi.org/10.1007/978-94-009-6000-8

31. *Ji S., Pan S., Cambria E., Marttinen P., Yu P.S.* A Survey on Knowledge Graphs: Representation, Acquisition, and Applications // IEEE Transactions on Neural Networks and Learning Systems. 2022. Vol. 33, No. 2. P. 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

32. *Kirillovich A.V., Nevzorova O.A., Lipachev E.K.* OntoMathPRO 2.0 Ontology: Up-dates of Formal Model // Lobachevskii J. of Math. 2022. Vol. 43, No. 12. P. 3504–3514. https://doi.org/10.1134/S1995080222150136

33. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A.* New components of the OntoMathPRO ontology for representing math knowledge // Nauchnyj servis v seti Internet. 2023. № 25. S. 141–151. https://doi.org/10.20948/abrau-2023-32

34. RDF Mapping Language (RML). Unofficial Draft, 20 June 2024. URL: https://rml.io/specs/rml/ (Accessed: 22.03.2026)

35. *Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A.* Digital Ecosystem OntoMath as an Approach to Building the Space of Mathematical Knowledge // Russian Digital Libraries Journal. 2023. Vol. 26, No. 2. P. 154–202. https://doi.org/10.26907/1562-5419-2023-26-2-154-202

36. *Elizarov A.M., Lipachev E.K.* Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge // Scientific and Technical Information Processing. 2023. Vol. 50, No. 1. P. 35–39. https://doi.org/10.3103/s0147688223010021

37. *Elizarov A., Lipachev E.* Big math methods in Lobachevskii-DML // CEUR Workshop Proc. 2019. Vol. 2523. P. 59–72. https://ceur-ws.org/Vol-2523/invited08.pdf (Accessed: 22.03.2026).

СВЕДЕНИЯ ОБ АВТОРАХ

ЛИПАЧЕВ Евгений Константинович – кандидат физико-математических наук, доцент кафедры цифровой аналитики и технологий искусственного интеллекта Института информационных технологий и интеллектуальных систем Казанского федерального университета, доцент Университета Иннополис. Научные интересы: цифровые библиотеки, интеллектуальный анализ данных, рекомендательные системы, технологии извлечения знаний.

Evgeny Konstantinovich LIPACHEV – Candidate of Physics and Mathematics, Associate Professor, Kazan Federal University, Innopolis University. Research interests: digital libraries, data mining, recommender systems, knowledge extraction technologies.

email: elipachev@gmail.com;

ORCID: 0000-0001-7789-2332

МУРАДЫМОВ Булат Русланович – магистрант Института информационных технологий и интеллектуальных систем Казанского федерального университета. Научные интересы: интеллектуальный анализ данных, рекомендательные системы, технологии извлечения знаний, графы знаний.

Bulat MURADYMOV – student at the Institute of Information Technologies and Intelligent Systems, Kazan Federal University. Research interests: data mining, recommender systems, knowledge extraction technologies, knowledge graphs.

email: muradymov.bulat@mail.ru;

ORCID: 0009-0004-1187-8158

Материал поступил в редакцию 23 марта 2026 года

УДК 004

ОНТОЛОГИЧЕСКИЙ ПОДХОД К ПРОЕКТИРОВАНИЮ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ

Е. А. Малых[1] [0009-0008-0730-1603], А. А. Блощук[2] [0000-0001-6683-5973], О. М. Атаева[3] [0000-0003-0367-5575]

[1–3]*Московский университет имени С. Ю. Витте, г. Москва, Россия*

[3]*Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия*

[1]warior227@yandex.ru, [2]abloshuk@muiv.ru, [3]oataeva@frccsc.ru

Аннотация

Несмотря на широкое использование микросервисной архитектуры в разработке программных систем, в настоящее время не существует формализованного подхода, обеспечивающего согласованное и гарантированное взаимодействие микросервисов на уровне передаваемых данных, что приводит к возникновению интеграционных ошибок и усложняет сопровождение распределенных систем. В работе предложен подход к организации взаимодействия микросервисов на основе онтологического моделирования, обеспечивающего формализацию структур данных и автоматизированную валидацию сообщений. Предложен метод преобразования в онтологических моделей формальных описаний схем данных основанный на спецификации схем GraphQL. Он позволяет автоматизировать процесс валидации данных и снизить количество интеграционных ошибок. Разработана также онтологическая модель, обеспечивающая анализ зависимостей между микросервисами и механизм валидации контрактов сообщений.

Практическая значимость работы заключается в достижении согласованного описания микросервисов, операций и форматов сообщений в результате использования онтологического подхода. Представление онтологии в виде графа позволяет анализировать зависимости между микросервисами и упрощает сопровождение крупных распределенных систем.

***Ключевые слова**: онтология, GraphQL Schema, интеграция данных, микросервисная архитектура, потоки сообщений, валидация данных, межсервисное взаимодействие, онтологическая модель, согласованность данных, управление схемами, шина данных.*

ВВЕДЕНИЕ

На сегодняшний день микросервисная архитектура является наиболее приоритетным выбором в проектировании распределенных программных систем. В них функционал разделен на отдельные автономные модули, называемые микросервисами. Они могут разрабатываться и использоваться независимо друг от друга. Актуальной проблемой является отсутствие согласованного обмена данными между микросервисами. При эволюции сервисов формат сообщений меняется, что приводит к интеграционным ошибкам. Нами рассмотрен подход к проектированию онтологической модели описания всей программной системы, который предусматривает взаимодействие на уровне абстракции, без привязки к конкретным языкам интеграции [1]. Такая онтология позволяет автоматически конвертировать модель в формальную схему и механизмы валидации. Практическое применение показано на примере онтологии в рамках информационной системы анализа научных текстов с целью их дальнейшей обработки.

БЛИЗКИЕ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

В работе [2] рассмотрена технология удаленного вызова процедур (Remote Procedure Call, RPC) как способ валидации сообщений. Авторы проанализировали существующие технологии с позиций повышения эффективности и производительности. Основное внимание уделено эффективности метода и формализации структуры данных преимущественно в контексте технологии Protocol Buffers. Преимуществами этого подхода являются строгая типизация и производительность, недостатком – привязка к конкретной технологии. Не учитываются также семантика и межсервисные зависимости, нет описания архитектуры системы.

В [3] исследована проблема согласованного управления данными в микросервисной системе, которая состоит из нескольких десятков сервисов. Автор

предложил использовать онтологию как инструмент описания метаданных. Предложенная модель позволяет абстрагироваться от каких-либо реализаций СУБД и накладывает ограничения на изменение данных микросервисами. На основе этой модели сформирован единый формат данных, согласованный между микросервисами. Но все же эта модель не является формализацией взаимодействия, описанием операций, потоков сообщений или же каких-либо контрактов сообщений. При этом, наглядно продемонстрировано применение онтологического подхода для решения задачи унификации и управления данными в микросервисной архитектуре.

Таким образом, существующие подходы, как правило, ориентированы на формализацию структур данных и не охватывают архитектурные аспекты взаимодействия микросервисов. Ограничения существующих решений указывают на необходимость разработки альтернативного подхода.

МОДЕЛЬ

В большинстве случаев взаимодействие между сервисами описывают фрагментно, причем API и форматы сообщений отдельно. Зависимости и полные цепочки взаимодействий между сервисами в каком-либо формализованном виде вовсе отсутствуют [4, 5]. В результате такого проектирования нет единой модели, которая позволила бы надежно управлять не только структурой самих данных, но и архитектурой сторонней системы, а именно нет ответа, какой сервис с каким может или должен взаимодействовать.

Нами была спроектирована онтология в редакторе Protégé, которая представляет собой единую концептуальную модель взаимодействия микросервисной архитектуры [6]. В ней описаны базовые сущности микросервиса (см. рис. 1), а также взаимодействие в виде операций и сообщений. На рисунке для облегчения понимания показан упрощенный вид спроектированной онтологии, представленной в формате PlantUML-схемы. Класс «Сервис» может предоставлять внешние методы взаимодействия в виде класса «API». Показана связь между классом API и классом операции, которая является самим методом API. Кроме того в формате сущностей описаны сообщения, схемы сообщений, поля схемы, а также тип поля. Указаны достаточно подробные связи между всеми сущностями.

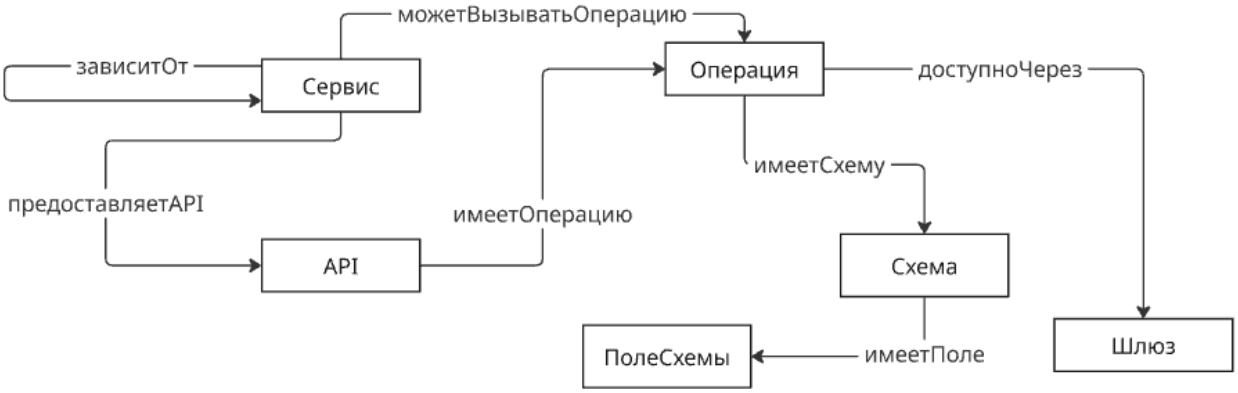


Рис. 1. Упрощенный вид спроектированной онтологии взаимодействия микросервиса.

Выбор онтологического подхода обусловлен его способностью объединять описания структуры данных и семантики взаимодействий в рамках единой модели [7]. По сравнению с существующими решениями предлагаемый онтологический метод является более сложным и затратным на начальном этапе проектирования, что связано с необходимостью построения и сопровождения онтологической модели. Но эти затраты компенсируются повышением согласованности архитектурных решений, а также обеспечивается централизованное управление схемами сообщений. Дополнительным преимуществом является формальный анализ межсервисных зависимостей. Такой анализ упрощает эволюцию схем без нарушения целостности архитектуры.

С точки зрения производительности дополнительные накладные расходы отсутствуют. Проверка сообщений выполняется на основе ограничений, заранее сгенерированных в формате GraphQL Schema. Обращение к онтологической модели при обработке сообщений не требуется. Основными задачами этого процесса являются обеспечение контроля жизненного цикла [8] всех микросервисов и проверка сообщений по согласованной схеме [9].

ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ

Для подтверждения практической применимости подхода была отдельно сформирована часть микросервисной системы, описанной в рамках разработанной онтологии. Рассматриваемый фрагмент включает три микросервиса: сервис

формирования эмбеддингов документов (EmbeddingService), сервис построения графов знаний (GraphService) и сервис загрузки (UploadService).

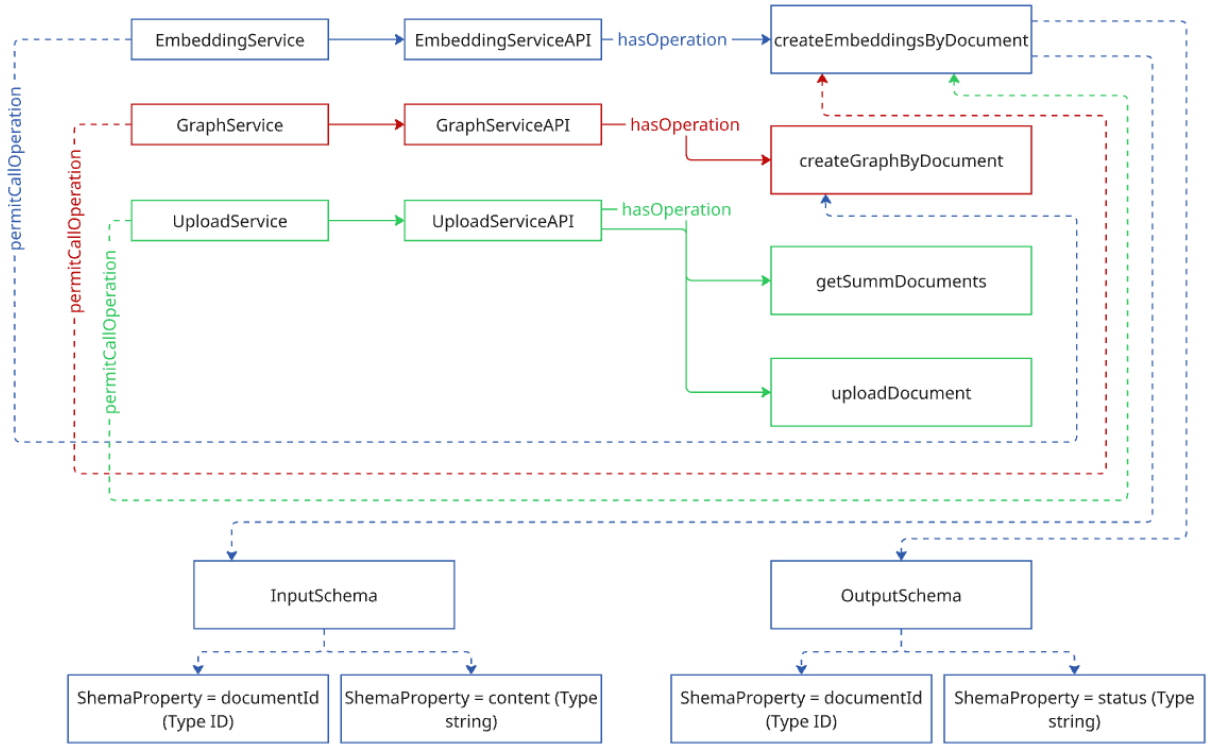


Рис. 2. Онтологическое представление сервисов, API-операций и правил их взаимодействия.

Диаграмма на рис. 2 отражает структуру сервисов, их программные интерфейсы, операции, а также схемы входных и выходных данных. Кроме того, на ней показаны правила допустимых вызовов операций между различными сервисами. Цвет элементов используется для обозначения принадлежности к сервису.

Сервис, обозначенный синим цветом, отвечает за обработку эмбеддингов документов. Он представлен компонентом EmbeddingService и интерфейсом EmbeddingServiceAPI. В интерфейсе определена операция createEmbeddingsByDocument. Она формирует векторное представление документа. Красным цветом обозначено все, что связано с сервисом построения графовой структуры знаний. Он представлен компонентом GraphService и интерфейсом GraphServiceAPI. Через этот интерфейс выполняется операция createGraphByDocument. Она формирует граф знаний на основе содержимого

документа. Зеленым цветом отмечен сервис работы с документами, содержащий компонент UploadService и интерфейс UploadServiceAPI. В интерфейсе определены операции uploadDocument и getSummDocuments. Первая выполняет загрузку документа, вторая возвращает сводную информацию о документах.

Связь между интерфейсами и операциями обозначена отношением hasOperation. Оно показывает, что интерфейс содержит определенные операции. Таким образом фиксируется структура сервиса, его интерфейса и доступных функций.

Пунктирные линии на диаграмме обозначают правила вызова операций между сервисами. Они связаны отношением permitCallOperation определяющим какие операции могут вызывать другие операции системы. Каждая пунктирная линия имеет цвет соответствующего сервиса, что показывает, какой сервис инициирует вызовы. В итоге устанавливаются зафиксированы допустимые взаимодействия между компонентами системы.

В нижней части диаграммы представлены схемы данных. Они описывают структуру входных и выходных параметров операций. Входная схема обозначена как InputSchema, она содержит свойства documentId и content. Идентификатор документа имеет тип ID. Содержимое документа имеет строковый тип. Выходная схема обозначена как OutputSchema, она содержит свойства documentId и status. Идентификатор использован для связи результата с исходным документом. Поле status отражает состояние выполнения операции.

Онтология описывает структуру сервисов, операции, типы данных и правила взаимодействия. Однако сама по себе модель не обеспечивает контроль выполнения этих правил во время работы системы. Для этого требуется механизм, который реализует заданные ограничения при обработке сообщений. В рассматриваемом подходе таким интерфейсом выступает GraphQL-сервер. Он принимает запросы и перенаправляет их к соответствующим сервисам. На этом этапе важно проверить корректность сообщения и допустимость выполнения операции. Проверка должна учитывать несколько условий. Сообщение должно соответствовать структуре GraphQL-схемы, а также необходимо убедиться, что вызываемая операция разрешена и соответствует правилам взаимодействия сервисов.

Механизм формирования ограничений

Возникает необходимость разработки механизма валидации сообщений и проверки разрешений операций. Такой механизм должен использовать информацию из онтологии и применять ее при обработке GraphQL-запросов. Это позволит обеспечить контроль корректности взаимодействия сервисов и соблюдение архитектурных ограничений системы.

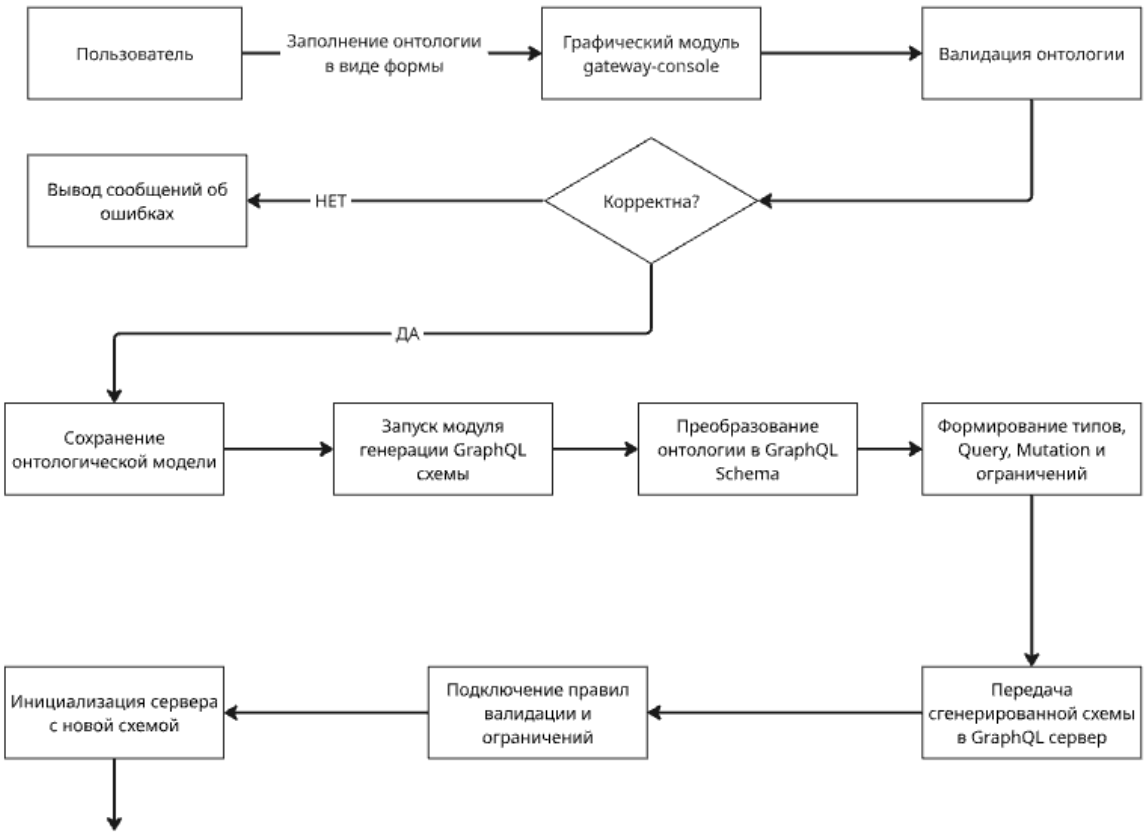


Рис. 3. Алгоритм формирования и применения ограничений GraphQL-сервера на основе онтологической модели.

На рис. 3 представлен алгоритм формирования ограничений для GraphQL-сервера на основе онтологической модели. Диаграмма показывает последовательность действий от ввода данных пользователем до инициализации сервера с новой схемой [10]. Процесс начинается с пользователя, который работает со специально разработанным графическим интерфейсом системы. Через форму он вводит или меняет элементы онтологии. Эти данные передаются в графический модуль gateway-console, использующийся для редактирования структуры модели и управления ее параметрами.

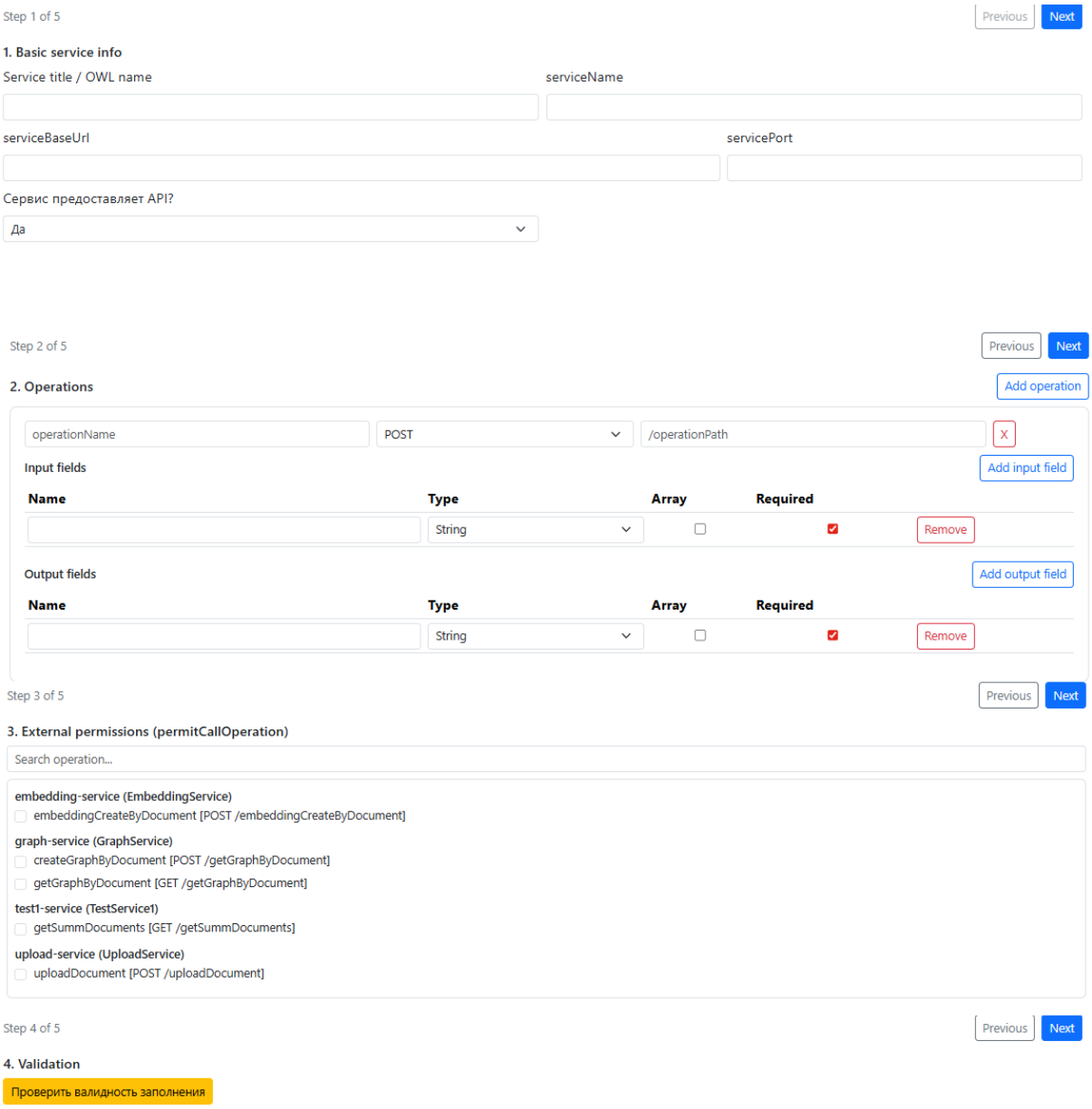


Рис. 4. Интерфейс пошагового заполнения параметров микросервиса и его операций в модуле формирования онтологии.

Интерфейс реализован в виде последовательности шагов и продемонстрирован на рис. 4. На первом этапе задаются основные параметры сервиса, включая его имя, адрес и порт. Далее пользователь описывает операции сервиса, указывает путь вызова, метод запроса и структуру входных и выходных данных. Для каждого поля определяются тип данных, обязательность и возможность использования массива. Следующий этап предназначен для задания разрешенных вызовов внешних операций. Пользователь может выбрать операции

других микросервисов, к которым разрешен доступ. Эти правила формируют ограничения взаимодействия между сервисами. На завершающем этапе выполняется проверка корректности введенных данных.

Система анализирует корректность структуры и связей между элементами. Результат проверки определяется условием корректности модели. Если обнаружены ошибки, система формирует сообщения об ошибках. Эти сообщения передаются пользователю. После получения сообщений об ошибках пользователь может исправить введенные данные. Если онтологическая модель проходит проверку, она сохраняется. Сохраненная структура используется в дальнейшем процессе генерации схемы. После этого запускается модуль генерации GraphQL-схемы. На этом этапе онтология преобразуется в формальное описание GraphQL Schema.

В результате преобразования формируются основные элементы схемы. Создаются типы данных, а также операции `Query` и `Mutation`. Одновременно формируются ограничения, которые будут применяться при обработке запросов.

На рис. 5 показан интерфейс формы этапа подтверждения создания сущности микросервиса. После заполнения формы пользователем выполняется автоматическая проверка корректности введенных данных. Если структура модели соответствует заданным ограничениям, система отображает окно подтверждения создания онтологических сущностей.

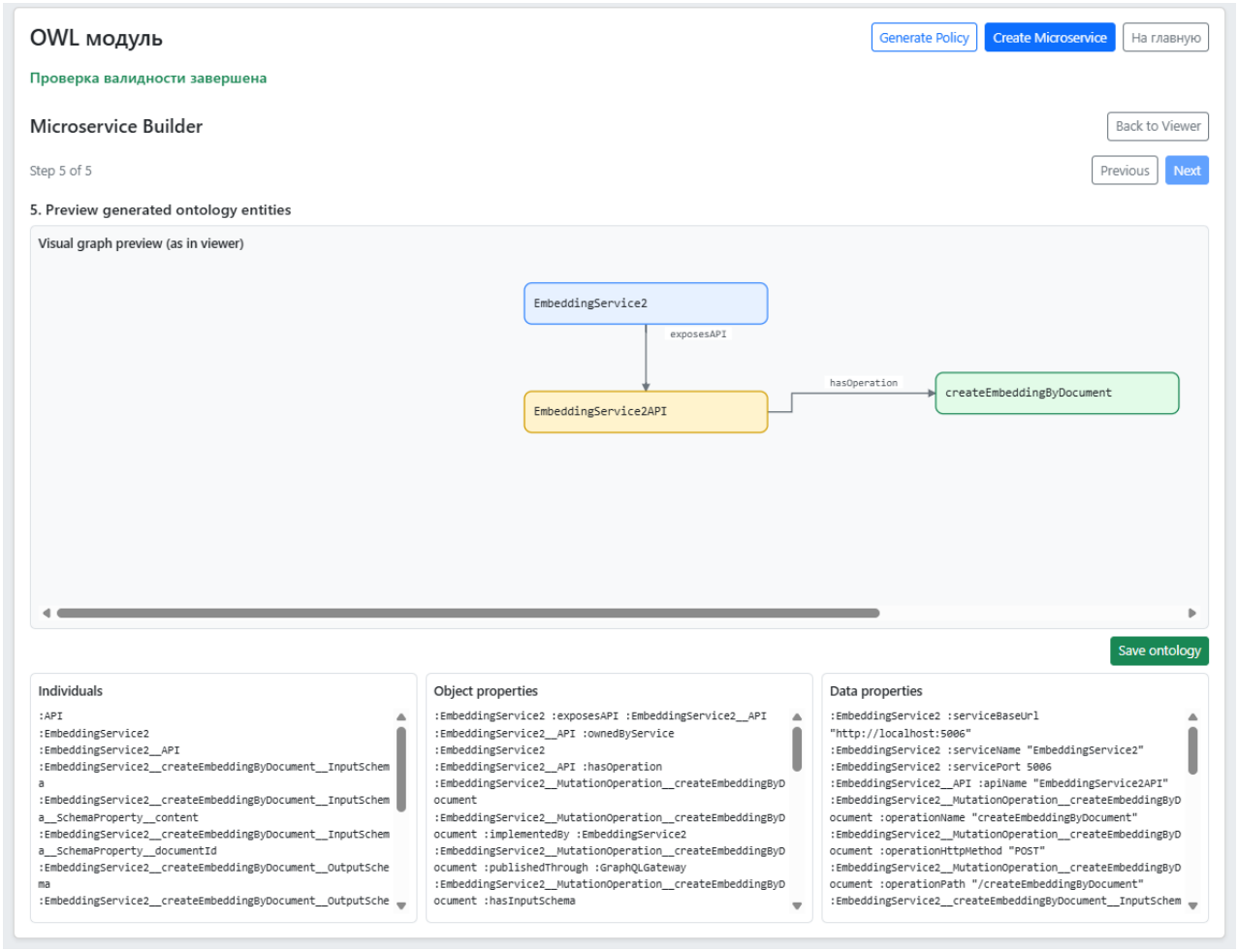


Рис. 5. Интерфейс предварительного просмотра и подтверждения созданных сущностей онтологической модели микросервиса.

В верхней части интерфейса отображается сообщение о завершении проверки корректности модели. Это означает, что структура онтологии успешно прошла валидацию. Пользователю доступен режим предварительного просмотра сформированных сущностей. Центральная часть окна содержит графическое представление онтологии. В ней визуализируются созданные элементы модели и связи между ними. В данном примере это сервис `EmbeddingService2`, его программный интерфейс `EmbeddingService2API` и операция `createEmbeddingByDocument`. Между элементами показываются отношения, которые описывают структуру сервиса и его API. Нижняя часть окна содержит текстовое представление созданных элементов онтологии, таких как индивидуумы, объектные свойства и свойства данных. Эти элементы отражают структуру

сервисов, их интерфейсов и операций, а также параметры конфигурации и описания методов.

Сформированная схема передается в GraphQL-сервер. Затем подключаются правила валидации и ограничения, полученные из онтологической модели. На завершающем этапе выполняется инициализация сервера с новой схемой. Это позволяет серверу использовать заданные правила при обработке входящих сообщений.

Механизм соблюдения ограничений

Механизм проверки входящих сообщений в GraphQL-сервере обеспечивает контроль и соблюдение заданных ограничений во время работы системы. Он анализирует структуру запроса и сопоставляет ее с описанием, полученным из онтологической модели. Процесс работы данного механизма представлен на рис. 6.

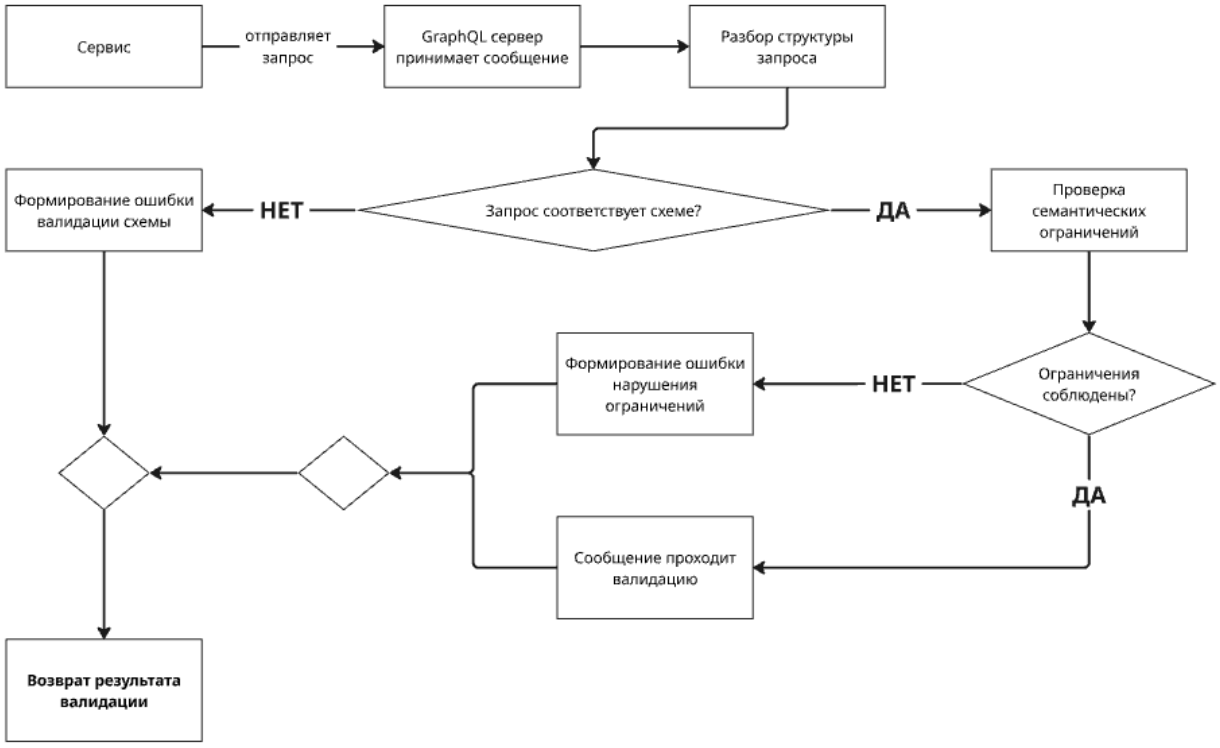


Рис. 6. Алгоритм валидации сообщений и проверки ограничений в GraphQL-сервере.

На рис. 6 представлена схема обработки запроса, поступающего в GraphQL-сервер. Процесс начинается с отправки запроса сервисом или клиентским приложением. Сообщение поступает на сервер, далее выполняется анализ структуры запроса. На этом этапе определяется соответствие запроса GraphQL-схеме, которая была ранее сгенерирована на основе онтологии. Если структура запроса не соответствует схеме, формируется сообщение об ошибке валидации. В этом случае дальнейшая обработка запроса не выполняется: сервер возвращает результат проверки с описанием ошибки. Если структура запроса корректна, выполняется следующий этап проверки. На этом этапе анализируются семантические ограничения: проверяются допустимость вызова операции и соответствие запроса правилам взаимодействия сервисов. Эти правила были определены ранее в онтологической модели.

В случае, когда все проверки выполнены успешно, сообщение считается корректным. Запрос проходит валидацию и может быть передан для дальнейшего выполнения. Проверка выполняется непосредственно в GraphQL-сервере. Это позволяет предотвращать выполнение недопустимых операций и поддерживать согласованность взаимодействия сервисов системы.

На рис. 7 приведен фрагмент кода, который демонстрирует результат преобразования экземпляров онтологии в GraphQL-схему. Этот код не отражает всю реализацию сервера. Он показывает только часть, связанную с ограничениями вызовов операций и проверкой структуры сообщений.

Названные элементы формируются автоматически на основе информации, содержащейся в онтологии. Фрагмент демонстрирует пример формирования правил разрешенных операций. В онтологической модели такие ограничения задаются через отношение разрешенных вызовов между сервисами. После конвертации эти данные преобразуются в структуру конфигурации GraphQL-сервера. В коде формируется объект `POLICY_RULES`. Он содержит описание допустимых запросов для каждого микросервиса. Для каждого сервиса указываются разрешенные операции `Query` и `Mutation`. В результате ограничения, заданные в онтологии, становятся исполняемыми правилами и учитываются при обработке запросов.

```
// From OWL ontology.
// Source: Microservice.serviceName + Microservice.permitCallOperation + operations
const POLICY_RULES = {
  "embedding-service": {
    Query: [
      "getGraphByDocument",
    ],
    Mutation: [
      "createGraphByDocument",
      "embeddingCreateByDocument",
    ],
  },
  "graph-service": {
    Query: [
      "getGraphByDocument",
      "getSummDocuments",
    ],
    Mutation: [
      "createGraphByDocument",
      "embeddingCreateByDocument",
    ],
  },
  "upload-service": {
    Query: [
      "getGraphByDocument",
    ],
    Mutation: [
      "createGraphByDocument",
      "embeddingCreateByDocument",
      "uploadDocument",
    ],
  },
};

module.exports = {
  POLICY_RULES,
};
```

Рис. 7. Фрагмент конфигурации ограничений вызова операций микросервисов, сформированный на основе онтологической модели.

На рис. 8 показан пример формирования части GraphQL-схемы. Создается тип ответа операции `createEmbeddingByDocument`, описывается структура возвращаемого сообщения. Указываются поля `documentId` и `message`, а также их типы. Далее определяется описание операции `Mutation`. В нем задаются параметры запроса и их типизация. Такой фрагмент представляет преобразование схем сообщений из онтологической модели в формальное описание GraphQL.

```
const embeddingServiceEmbeddingcreatebydocumentType = new GraphQLObjectType({
  name: "EmbeddingServiceEmbeddingcreatebydocument",
  fields: () => ({
    documentId: { type: new GraphQLNonNull(GraphQLID) },
    message: { type: new GraphQLNonNull(GraphQLString) },
  }),
});

const embeddingServiceQueryFields = {
};

const embeddingServiceMutationFields = {
  embeddingCreateByDocument: {
    type: embeddingServiceEmbeddingcreatebydocumentType,
    args: {
      content: { type: new GraphQLNonNull(GraphQLString) },
      documentId: { type: new GraphQLNonNull(GraphQLID) },
    },
```

Рис. 8. Фрагмент GraphQL схемы операции, сформированной на основе схемы сообщения из онтологии.

ЗАКЛЮЧЕНИЕ

Разработанная онтология микросервисов является единой формальной моделью архитектуры всей системы. Под онтологией микросервисов понимается формальная семантическая модель, описывающая сущности микросервисной архитектуры и их взаимосвязи. Особенностью онтологического метода проектирования является автоматическая согласованность всей цепочки взаимодействия. Поскольку онтология содержит не только сущности, но и связи между ними, после загрузки в графовую базу можно выполнять сложные запросы, например поиск всех сервисов, зависящих от какого-либо события [11]. Микросервисы, операции, сообщения и схемы данных становятся вершинами графа, а их взаимодействие и зависимости – ребрами. Графовое представление [12] позволяет организовать извлечение информации и генерацию модели взаимодействия, ориентированные на архитектуру системы. В дальнейщем возможна разработка интеллектуального ассистента [13]. Он будет способен отвечать на вопросы о структуре системы, доступных операциях, форматах сообщений. Это существенно снизит порог вхождения для новых разработчиков системы. Предлагаемый подход снижает время изучения документации и устраняет необходимость ручного анализа API и схем сообщений. Это особенно важно для крупных микросервисных систем, где количество сервисов и интеграций может доходить до нескольких десятков, а то и сотен.

СПИСОК ЛИТЕРАТУРЫ

1. *Oumoussa I., Saidi R.* The ontology-based mapping of microservice identification approaches: a systematic study of migration strategies from monolithic to microservice architectures // Computers. 2025. Vol. 14, No. 4. P. 133.

2. Шитько А.М., Пацей Н.В. Интеграция микросервисов на основе RPC // Информационные технологии: тезисы докладов 82-й научно-технической конференции профессорско-преподавательского состава, научных сотрудников и аспирантов (с международным участием). Минск: Белорусский государственный технологический университет, 2018. С. 31–32.

3. *Балес А.И.* Унифицированная модель данных и ее применение в микросервисной архитектуре // Современные информационные технологии и ИТ-образование. 2020. Т. 16, № 2. С. 416–425.
https://doi.org/10.25559/SITITO.16.202002.416-425

4. *Anderson C. et al*. An ontology-based reasoning framework for context-aware applications // Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context. Cham: Springer International Publishing, 2015. P. 471–476.

5. *Гусенков А.М., Бухараев Н.Р., Биряльцев Е.В.* Построение онтологии предметной области на основе логической модели данных // Электронные библиотеки. 2020. Т. 23, № 3. С. 390–417.
https://doi.org/10.26907/1562-5419-2020-23-3-390-417

6. *Карев А.Н., Федосин С.А.* Онтологический подход к интеграции информационных систем // Перспективы науки. 2023. Т. 168, № 9. С. 26–29.

7. *Чернов П.К., Рабчевский Е.А.* Создание интегрированной модели данных из разнородных источников, содержащих цифровые следы // Вестник ПГУ. Математика. Механика. Информатика. 2022. С. 81–87.
https://doi.org/10.17072/1993-0550-2022-2-81-87

8. *Ketul Kishorbhai Dusane* Cloud Messaging Systems Architecture and Implementation // Journal of Computer Science and Technology Studies. 2025. Vol. 7, No. 8. P. 739–746. https://doi.org/10.32996/jcsts.2025.7.8.86

9. Huanyu Li, Olaf Hartig, Rickard Armiento, Patrick Lambrix. Ontology-Based GraphQL Server Generation for Data Access and Data Integration // Semantic Web. 2024. Vol. 15, No. 5. P. 1639–1675.

10. *Ломов П.А., Малоземова М.Л.* Обучение и применение нейросетевой языковой модели для пополнения онтологии // Труды Кольского научного центра РАН. 2020. № 8–11. С. 38–45.

11. *Зимнуров М.Ф., Астраханцева И.А.* Методология создания многосвязных структур данных с применением LLM в рабочих проектах // Современные наукоемкие технологии. Региональное приложение. 2025. № 1. С. 76–83.

12. *Папуша С.И.* Онтология и графовые базы данных // Проблемы экономики и юридической практики. 2020. Т. 16, № 3. С. 268–272.

13. *Ломов П.А., Шишаев М.Г., Диковицкий В.В.* Преобразование OWL-онтологий для визуализации и использования в качестве основы пользовательского интерфейса // Онтология проектирования. 2012. № 3. С. 49–61.

AN ONTOLOGICAL APPROACH TO DESIGNING A MICROSERVICE ARCHITECTURE

E. A. Malykh[1] [0009-0008-0730-1603], A. A. Bloshchuk[2] [0000-0001-6683-5973], O. M. Ataeva[3] [0000-0003-0367-5575]

[1–3]*Moscow Witte University, Moscow, Russia*

[3]*Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia*

[1]warior227@yandex.ru, [2]abloshuk@muiv.ru, [3]oataeva@frccsc.ru

Abstract

Despite the widespread use of microservice architecture in the development of software systems, there is no formalized approach that ensures consistent and guaranteed interaction of microservices at the level of transmitted data, which leads to integration errors and complicates the maintenance of distributed systems. The purpose of the study is to develop an approach to the organization of microservices interaction based on ontological modeling, providing formalization of data structures and automated validation of messages. The paper presents a method for converting formal descriptions of data schemas into ontological models based on the GraphQL schema specification. This method allows you to automate the data validation process

and reduce the number of integration errors. An ontological model has been developed that provides an analysis of dependencies between microservices and a mechanism for validating message contracts.

The practical significance of the work lies in achieving a consistent description of microservices, operations, and message formats as a result of using an ontological approach. The representation of the ontology in the form of a graph makes it possible to analyze the dependencies between microservices and simplifies the maintenance of large distributed systems.

Keywords*: ontology, GraphQL schema, data integration, microservice architecture, message flows, data validation, service interoperability, ontological model, data consistency, schema management, data bus.*

REFERENCES

1. *Oumoussa I., Saidi R.* The ontology-based mapping of microservice identification approaches: a systematic study of migration strategies from monolithic to microservice architectures // Computers. 2025. Vol. 14, No. 4. P. 133.

2. *Shitko A.M., Patsei N.V.* Integration of microservices based on RPC // Information Technologies: Proceedings of the 82nd Scientific and Technical Conference of Academic Staff, Researchers, and Postgraduate Students (with international participation). Minsk: Belarusian State Technological University, 2018. P. 31–32.

3. *Bales A.I.* A unified data model and its application in microservice architecture // Modern Information Technologies and IT Education. 2020. Vol. 16, No. 2. P. 416–425. https://doi.org/10.25559/SITITO.16.202002.416-425

4. *Anderson C. et al.* An ontology-based reasoning framework for context-aware applications // Proceedings of the International and Interdisciplinary Conference on Modeling and Using Context. Cham: Springer International Publishing, 2015. P. 471–476.

5. *Guseenkov A.M., Bukharaev N.R., Biryaltsev E.V.* Construction of a domain ontology based on a logical data model // Russian Digital Library. 2020. Vol. 23, No. 3. P. 390–417. https://doi.org/10.26907/1562-5419-2020-23-3-390-417

6. *Karev A.N., Fedosin S.A.* An ontological approach to the integration of information systems // Science Prospects. 2023. Vol. 168, No. 9. P. 26–29.

7. *Chernov P.K., Rabchevsky E.A.* Creation of an integrated data model from

heterogeneous sources containing digital traces // Bulletin of PSU. Mathematics. Mechanics. Informatics. 2022. P. 81–87.
https://doi.org/10.17072/1993-0550-2022-2-81-87

8. *Dusane K.K.* Cloud messaging systems architecture and implementation // Journal of Computer Science and Technology Studies. 2025. Vol. 7, No. 8. P. 739–746. https://doi.org/10.32996/jcsts.2025.7.8.86

9. *Li H., Hartig O., Armiento R., Lambrix P.* Ontology-based GraphQL server generation for data access and data integration // Semantic Web. 2024. Vol. 15, No. 5. P. 1639–1675.

10. *Lomov P.A., Malozemova M.L.* Training and application of neural network language models for ontology population // Proceedings of the Kola Science Center of the Russian Academy of Sciences. 2020. No. 8–11. P. 38–45.

11. *Zimnurov M.F., Astrakhantseva I.A.* Methodology for creating multi-connected data structures using LLMs in practical projects // Modern High Technologies. Regional Application. 2025. No. 1. P. 76–83.

12. *Papusha S.I.* Ontologies and graph databases // Problems of Economics and Legal Practice. 2020. Vol. 16, No. 3. P. 268–272.

13. *Lomov P A., Shishaev M.G., Dikovitsky V.V.* Transformation of OWL ontologies for visualization and use as a basis for user interfaces // Ontology of Designing. 2012. No. 3. P. 49–61.

СВЕДЕНИЯ ОБ АВТОРАХ

МАЛЫХ Евгений Александрович – аспирант Московского университета имени С. Ю. Витте по направлению «Системный анализ, управление и обработка информации, статистика». Научные интересы: автоматизация бизнес-процессов, интеграция источников данных в разнородных информационных системах.

Evgeniy Aleksandrovich MALYKH – PhD student at Moscow Witte University, specializing in System Analysis, Management, and Information Processing, Statistics. Research interests: business process automation, integration of data sources in heterogeneous information systems.

email: warior227@yandex.ru;

ORCID: 0009-0008-0730-1603

БЛОЩУК Андрей Алексеевич – кандидат технических наук, доцент кафедры информационных систем, Московский университет имени С. Ю. Витте. В 2004 году присвоена степень кандидата технических наук в 4-м ВНИИ (г. Юбилейный, МО). Более десяти лет преподавания дисциплин ИТ-направленности. Издан ряд научных статей по инновационным методам автоматизации различных ас пектов деятельности образовательной организации. Сфера научных интересов: автоматизация процессов образовательной деятельности. Применение информационных технологий в образовании.

Andrey Alekseevich BLOSHCHUK – Candidate of Engineering Sciences (PhD equivalent), associate professor at the Department of Information Systems, Moscow Witte University. He was awarded his Candidate of Engineering Sciences degree in 2004 from the 4th Central Research Institute (4th VNII) in Yubileyny, Moscow Oblast. With over a decade of experience teaching IT-related disciplines, he has authored numerous scientific articles on innovative methods for automating various aspects of educational organization activities. His research interests include the automation of educational processes and the application of information technologies in education.

email: abloshuk@muiv.ru

ORCID: 0000-0001-6683-5973

АТАЕВА Ольга Муратовна – старший научный сотрудник Вычислительного центра им. А.А. Дородницына ФИЦ ИУ РАН, к. т. н. Область научных интересов: системное программирование, базы данных, и нженерия знаний и онтологии.

Olga Muratovna ATAEVA – senior researcher at the Dorodnicyn Computing Center, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences; Candidate of Technical Sciences (PhD). Research interests: systems programming, databases, knowledge engineering, ontologies. Number of scientific publications – 80.

email: oataeva@frccsc.ru

ORCID: 0000-0003-0367-5575

Материал поступил в редакцию 12 апреля 2026 года

УДК 004.8:620.9

ИНТЕГРАЦИЯ СЕМАНТИЧЕСКОГО МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ ДЛЯ АНАЛИЗА ПРОБЛЕМ ЭНЕРГЕТИЧЕСКОЙ БЕЗОПАСНОСТИ

А. Г. Массель[1] [0000-0002-0351-0415], Т. Г. Мамедов[2] [0000-0002-3396-5074]

[1, 2]*Институт систем энергетики им. Л.А. Мелентьева Сибирского отделения РАН, г. Иркутск, Россия*

[1]amassel@gmail.com, [2]mamedowtymur@yandex.ru

Аннотация

Рассмотрена задача интеграции когнитивного и математического моделирования в исследованиях направлений развития топливно-энергетического комплекса с учетом требований энергетической безопасности. Актуальность работы обусловлена тем, что в существующей двухуровневой технологии исследований переход от результатов качественного анализа с помощью когнитивного моделирования к параметрам математической модели в значительной степени выполняется вручную, что снижает воспроизводимость численных экспериментов и ограничивает эффективность использования накопленных знаний. Цель проведенного исследования состояла в разработке программного компонента, обеспечивающего совместное использование когнитивной и математической моделей в составе Экосистемы знаний в энергетике. Предложен программный компонент, реализованный в составе комплекса ИНТЭК-SAW и обеспечивающий преобразование изменений когнитивной модели в параметры экономико-математической модели, а также обратную интерпретацию результатов расчетов. Разработана технология проведения численного эксперимента, включающая построение семантических (онтологической и когнитивной) моделей, формирование вычислительного сценария, выполнение оптимизационных расчетов и представление результатов, отличающаяся автоматизацией совместного использования онтологических, когнитивных и экономико-математических моделей. Для учета неопределенности предложен численный метод стохастической корректировки параметров на основе когнитивных весов. Работоспособность подхода продемонстрирована на численном эксперименте по исследованию влияния

ограничений выбросов CO_2 на топливно-энергетические балансы Сибирского федерального округа. Практическая значимость работы состоит в повышении обоснованности и воспроизводимости исследований развития топливно-энергетического комплекса за счет согласованного использования средств качественного и количественного анализа.

***Ключевые слова**: топливно-энергетический комплекс, энергетическая безопасность, когнитивное моделирование, онтологии, численный эксперимент, линейное программирование.*

ВВЕДЕНИЕ

Исследования направлений развития топливно-энергетического комплекса с учетом требований энергетической безопасности относятся к числу сложных задач, требующих совместного использования качественных и количественных методов анализа. В Институте систем энергетики им. Л. А. Мелентьева Сибирского отделения Российской академии наук для решения таких задач применяется двухуровневый подход, в рамках которого методы семантического (онтологического и когнитивного) моделирования используются на этапе качественного анализа, а экономико-математические модели и программные комплексы для оптимизационных расчетов – на этапе количественного анализа вариантов развития топливно-энергетического комплекса (ТЭК) [1–3]. Такой подход позволяет учитывать как структурные и причинно-следственные связи между факторами, так и количественные последствия реализации различных возмущающих и управляющих воздействий.

Однако существующие инструментальные средства практически не формализованный переход от результатов когнитивного анализа к параметрам математической модели [4]. На практике изменение сценарных факторов, выделенных на когнитивной карте, и их перенос в параметры и ограничения экономико-математической модели во многих случаях выполняются вручную. Это снижает воспроизводимость численных экспериментов, затрудняет повторное использование знаний и ограничивает возможности сопоставления различных вариантов исследования в рамках единого программного комплекса. В этих условиях возникает задача разработки программного компонента, обеспечивающего интеграцию когнитивного и математического моделирования в составе

цифровой среды системных исследований в энергетике. Такой компонент должен поддерживать построение и редактирование когнитивных моделей, формализованное преобразование изменений их весов и характеристик в параметры экономико-математической модели, выполнение расчетов по сформированным сценариям и обратную интерпретацию полученных результатов. Особое значение имеет также учет неопределенности, сопровождающей экспертные оценки силы и направленности воздействий при формировании сценариев развития ТЭК.

Целью проведенных исследований была разработка программного компонента интеграции когнитивного и математического моделирования в составе Экосистемы знаний в энергетике [1]. Для достижения поставленной цели решались задачи разработки архитектуры компонента в составе комплекса ИНТЭК-SAW, формализации технологии проведения численного эксперимента, разработки численного метода стохастической корректировки параметров на основе когнитивных весов и апробации предложенного подхода на примере исследования влияния ограничений выбросов CO_2 на параметры топливно-энергетического баланса Сибирского федерального округа.

В работе представлены архитектура программного компонента, технология проведения численного эксперимента, численный метод учета неопределенности и полученные результаты численного эксперимента.

1. ПРОГРАММНЫЙ КОМПЛЕКС ИНТЭК-SAW

Программный комплекс (ПК) ИНТЭК-SAW реализован на основе агентно-сервисной архитектуры, обеспечивающей согласованное использование сервисов семантического и математического моделирования, расчетных средств и средств представления результатов [5]. В рамках данной архитектуры отдельные функциональные задачи распределены между специализированными агентами, за счет этого обеспечено согласованное использование различных видов моделей и результатов вычислений в составе единого программного комплекса.

Архитектура ПК ИНТЭК-SAW (рис. 1) ориентирована на поддержку интеграции когнитивного и математического моделирования при проведении численных экспериментов по установлению направлений развития топливно-энергетического комплекса с учетом требований энергетической безопасности. В составе

комплекса выделены агенты формирования сценариев, построения и редактирования когнитивных моделей, а также выполнения оптимизационных расчетов и представления результатов исследования.

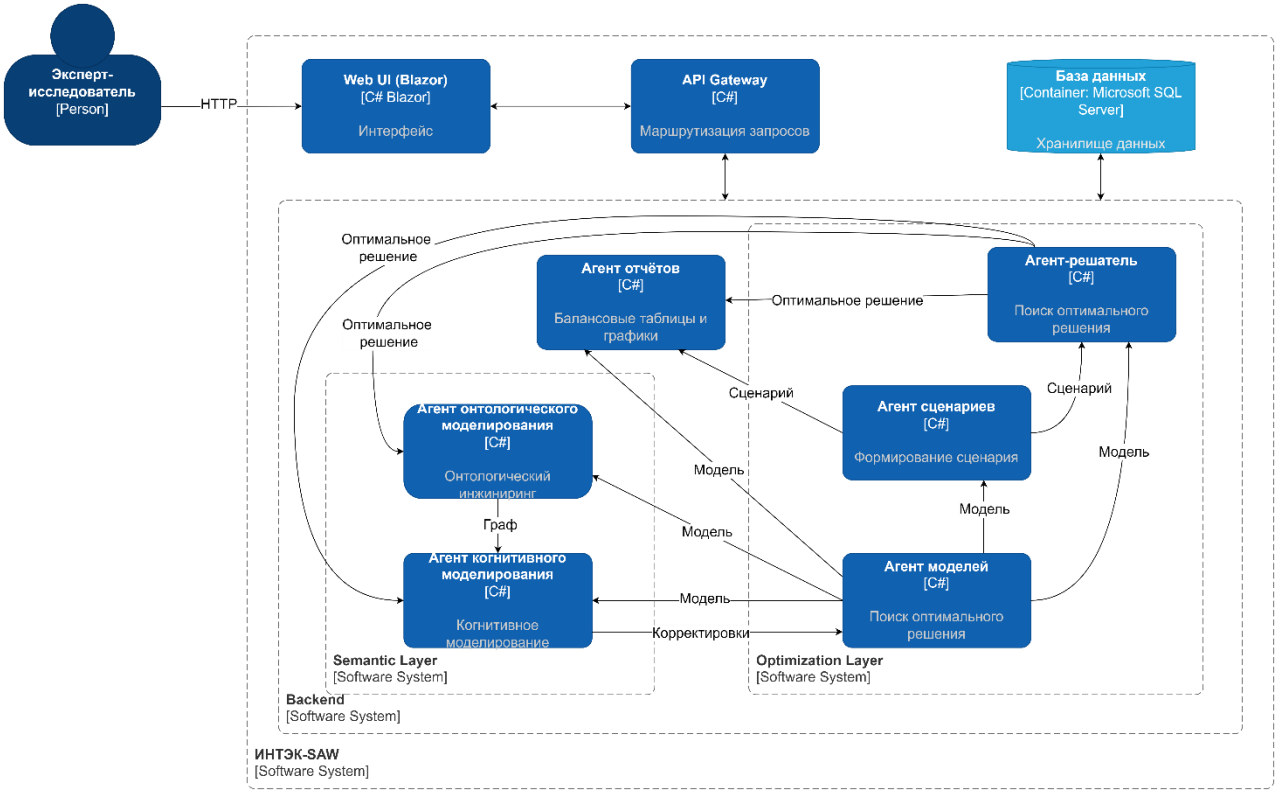


Рис. 1. Архитектура ПК ИНТЭК-SAW.

Агент когнитивного моделирования обеспечивает построение, редактирование и хранение когнитивных карт, отражающих причинно-следственные связи между факторами, параметрами и показателями исследуемой системы. Интерфейс агента представлен на рис. 2. Использование графового представления позволяет формализовать взаимосвязи между элементами модели и обеспечить их последующее преобразование в параметры экономико-математической модели. Тем самым программный комплекс ИНТЭК-SAW обеспечивает согласованное использование семантического и математического моделирования при проведении численного эксперимента.

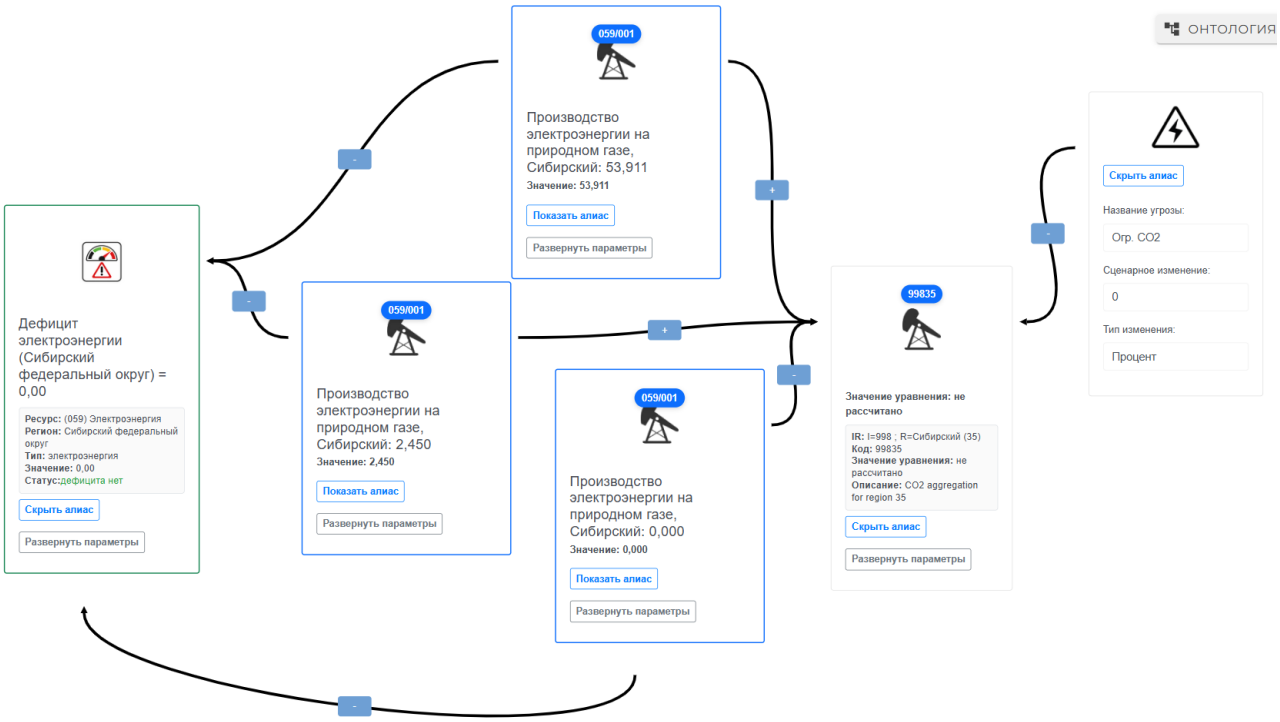


Рис. 2. Интерфейс построения и редактирования когнитивной карты (графического представления когнитивной модели)

2. ТЕХНОЛОГИЯ ПРОВЕДЕНИЯ ЧИСЛЕННОГО ЭКСПЕРИМЕНТА

Предложена технология проведения численного эксперимента с использованием агента когнитивного моделирования, включающая шесть основных этапов (рис. 3).

Этап 1. Выбирается модель для исследования, формируются технологические словари, задаются система переменных и ограничений, целевая функция. Результатом этапа является формализованная экономико-математическая модель ТЭК, представленная системой ограничений и целевой функцией [6]. Базовый вариант модели используется в качестве исходного состояния численного эксперимента.

Этап 2. Выполняется структурирование знаний о взаимосвязях между элементами экономико-математической модели. Эти отношения представляются в виде онтологии ориентированного графа, в котором вершины соответствуют факторам модели, а дуги отражают логико-семантические связи между ними. Результатом этапа является онтологическая модель ТЭК, формирующая семантический слой исследования [7].

Этап 3. На основе онтологической модели строится когнитивная карта, предназначенная для задания сценариев развития ТЭК. В нее включаются факторы модели, а также возмущающие и управляющие воздействия. Между ними задаются причинно-следственные связи в знаковой, весовой или функциональной формах. Результатом этапа является когнитивная модель, интегрированная с экономико-математической моделью через механизм преобразования изменений факторов когнитивной модели в параметры математической модели.

Этап 4. Формируется сценарий развития ТЭК с использованием когнитивной модели. Изменение значений факторов, характеризующих возмущающие или управляющие воздействия, приводит к корректировке параметров и ограничений технико-экономической модели. Результатом этапа является вычислительный сценарий, задающий согласованную совокупность условий функционирования ТЭК.

Этап 5. Выполняется поиск оптимального решения для каждого варианта модели ТЭК, входящего в вычислительный сценарий. Результатом этапа является построенный вариант развития ТЭК, характеризующийся количественными значениями переменных математической модели.

Этап 6. Дается интерпретация результатов численного эксперимента. Полученные данные представляются в виде балансовых таблиц и значений показателей на когнитивной карте. При необходимости результаты используются для уточнения исследуемых параметров ТЭК и повторного проведения расчета.

Таким образом, предложенная технология обеспечивает последовательный переход от формирования математической и семантической моделей к построению сценариев развития ТЭК, их количественному расчету и обратной интерпретации результатов в виде когнитивных карт [8].

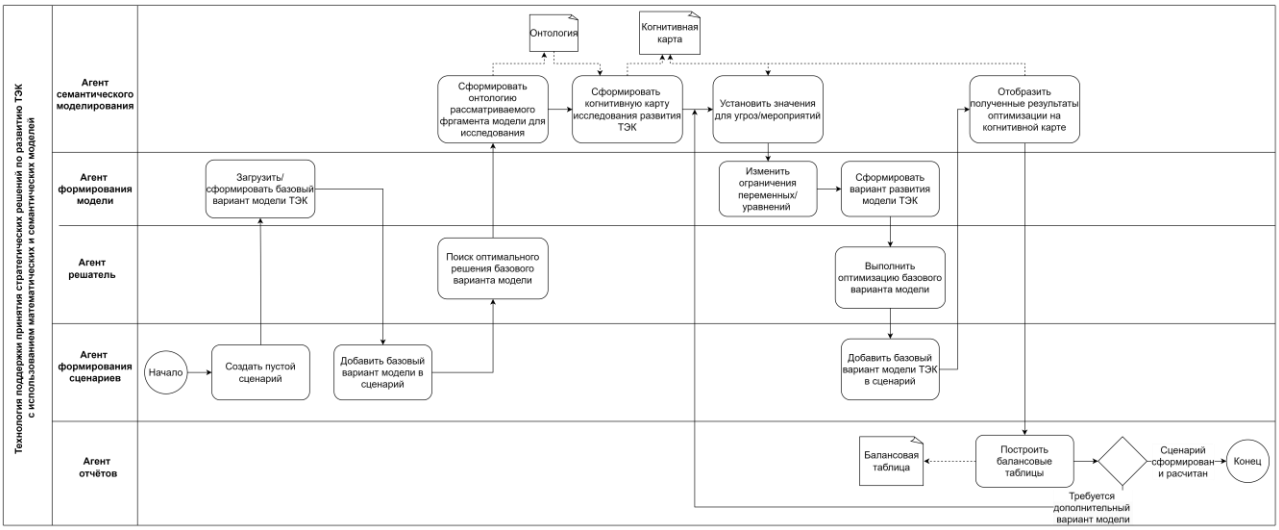


Рис. 3. Диаграмма бизнес-процесса технологии проведения численного эксперимента (BPMN).

Формирование вычислительного сценария на основе когнитивной модели требует учета неопределенности, связанной с оценкой силы воздействий и их влияния на параметры экономико-математической модели. Поэтому на данном этапе используется авторский численный метод стохастической корректировки параметров на основе когнитивных весов, обеспечивающий переход от детерминированного задания сценарных воздействий к формированию ансамбля возможных состояний ТЭК.

3. ЧИСЛЕННЫЙ МЕТОД СТОХАСТИЧЕСКОЙ КОРРЕКТИРОВКИ ПАРАМЕТРОВ НА ОСНОВЕ КОГНИТИВНЫХ ВЕСОВ

Для учета неопределенности в сценарных исследованиях ТЭК предложен численный метод стохастической корректировки параметров экономико-математической модели на основе когнитивных весов. В этом методе использована когнитивная карта [9], представляемая ориентированным взвешенным графом

$$G = (V_c, E_c),$$

где $V_c = \{v_1, v_2, \dots, v_n\}$ – множество концептов, а $E_c \subseteq V_c \times V_c$ – множество причинно-следственных связей между ними.

Каждая связь $(v_j \to v_i) \in E_c$ характеризуется весовым коэффициентом $w_{ji} \in [-1,1]$ и степенью уверенности эксперта [10] $c_{ji} \in [0,1]$. Для интеграции когнитивного и математического моделирования введено отображение

$\Phi: V_c \to P$, сопоставляющее концептам когнитивной карты параметры математической модели. Экономико-математическая модель ТЭК представлена в виде задачи оптимизации

$$\min_{x \in X} F(x, p),$$

где x – вектор переменных модели, $p = (p_1, p_2, \dots, p_m)$ – вектор параметров, $p_i^{(0)}$ – их базовые значения.

Для каждого концепта v_i рассмотрено множество входящих воздействий

$$\mathrm{In}(v_i) = \{(v_j \to v_i) \in E_c\}.$$

Агрегированное когнитивное воздействие определено как нормированная взвешенная сумма входящих влияний:

$$W_i = \sum_{(j \to i) \in In(v_i)} w_{ji}\, c_{ji} \Big/ \sum_{(j \to i) \in In(v_i)} c_{ji}$$

при условии, что $\sum_{(j \to i) \in In(v_i)} c_{ji} > 0$. Если для концепта v_i входящие связи отсутствуют, принимается $W_i = 0$. Такая форма позволяет учитывать как знак и силу когнитивного влияния, так и степень доверия к экспертным оценкам, не допуская искусственного увеличения суммарного эффекта только за счет числа входящих связей.

Суммарная степень уверенности экспертов определена выражением

$$C_i = 1 - \prod_{(j \to i) \in In(v_i)} (1 - c_{ji}).$$

Эта величина интерпретируется как агрегированная мера уверенности по совокупности входящих воздействий и принимает значения в интервале [0,1].

Стохастическая корректировка параметра p_i в k-й реализации моделируется случайной величиной

$$\Delta p_i^{(k)} \sim N(\mu_i, \sigma_i^2),$$

где

$$\mu_i = \beta W_i p_i^{(0)}, \quad \sigma_i = \alpha \mid p_i^{(0)} \mid (1 - C_i).$$

Здесь β – коэффициент, задающий масштаб среднего смещения параметра под действием когнитивного воздействия, α – коэффициент, определяющий уровень

стохастической вариации. Тогда значение параметра в k-й реализации находится в виде

$$p_i^{(k)} = \Pi_{\left[p_i^{\min},\, p_i^{\max}\right]}\left(p_i^{(0)} + \Delta p_i^{(k)}\right),$$

где Π – оператор проекции на допустимый интервал изменения параметра.

Для каждой реализации параметров $p^{(k)}$, $k = 1, \ldots, N$, решалась оптимизационная задача

$$x^{(k)} = \arg\min_{x \in X} F(x, p^{(k)}),$$

в результате формируется ансамбль сценарных состояний

$$V^{\mathrm{SCWA}} = \{V^{(k)}\}_{k=1}^{N}.$$

Агрегирование результатов численного эксперимента выполнено на основе статистических характеристик исследуемых индикаторов. Для индикатора I_r найдены оценки математического ожидания

$$E[I_r] = \frac{1}{N}\sum_{k=1}^{N} I_r^{(k)}$$

и дисперсии

$$\mathrm{Var}(I_r) = \frac{1}{N-1}\sum_{k=1}^{N}\left(I_r^{(k)} - E[I_r]\right)^2.$$

Предложенный метод обеспечивает формализацию связей когнитивной модели с параметрами экономико-математической модели и формирование ансамбля сценарных состояний ТЭК в условиях неопределенности.

4. ЧИСЛЕННЫЙ ЭКСПЕРИМЕНТ ПО СНИЖЕНИЮ ВЫБРОСОВ CO_2 В СИБИРСКОМ ФЕДЕРАЛЬНОМ ОКРУГЕ

Для апробации предложенной технологии и разработанного численного метода был проведен численный эксперимент по исследованию влияния ужесточения ограничений по выбросам CO_2 на параметры топливно-энергетического баланса Сибирского федерального округа.

В качестве базового значения экологического ограничения принималось верхнее ограничение уравнения «Выбросы CO_2 в Сибирском федеральном округе», с кодом 99835, равное 123462.74 тыс. т CO_2. В когнитивной модели возмущающему воздействию, отражающему ужесточение углеродной политики,

были сопоставлены значения весового коэффициента $w = -0.7$ и степени уверенности $c = 0.75$. Для стохастической корректировки параметра брались коэффициент масштаба $\beta = 0.035$ и коэффициент неопределенности $\alpha = 0.05$. На этой основе был сформирован ансамбль из 1000 расчетных реализаций, для каждой из которых выполнялся оптимизационный расчет модели ТЭК.

Выполненная серия расчетов позволила перейти от анализа одного детерминированного сценария к исследованию множества допустимых состояний системы при вариативном значении экологического ограничения. Несмотря на то что среднее смещение соответствует ужесточению ограничения, стохастическая постановка допускает реализацию значений параметра как ниже, так и выше базового уровня в пределах сформированного ансамбля. В полученном ансамбле значение ограничения по выбросам CO_2 изменялось в диапазоне от 116545.22 до 125669.92 тыс. т. Тем самым экологическое ограничение рассматривалось не как фиксированное условие, а как диапазон возможных состояний, формируемый за счет возмущающего воздействия и неопределенности экспертной оценки.

Результаты расчетов показали, что в большинстве реализаций устойчивость электроэнергетического баланса Сибирского федерального округа сохраняется (рис. 4). В основной части сценариев дефицит электроэнергии отсутствует, однако при наиболее жестких значениях ограничения возникают его ненулевые значения. По выборке максимальный дефицит составил 3.23 млрд кВт·ч; в ряде реализаций наблюдаются промежуточные значения порядка 0.05–2.88 млрд кВт·ч, тогда как в большинстве случаев значение дефицита остается нулевым. Это указывает на то, что исследуемая система в целом сохраняет работоспособность, а влияние ограничения проявляется прежде всего не в нарушении энергоснабжения, а в перестройке структуры потребления котельно-печного топлива.

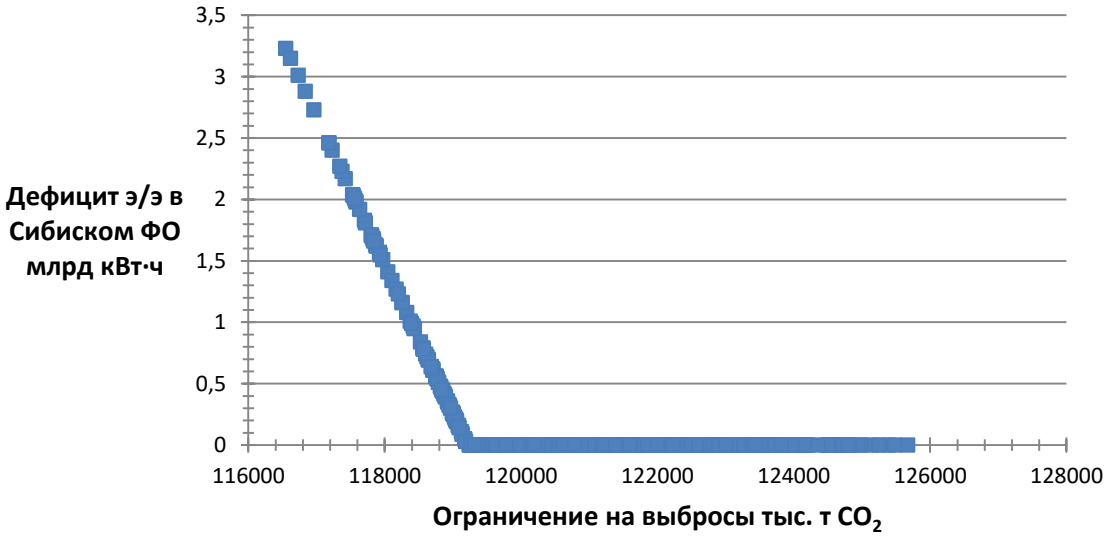


Рис. 4. Зависимость дефицита электроэнергии в Сибирском федеральном округе от уровня ограничения выбросов CO_2.

Анализ распределения показателей котельно-печного топлива подтвердил сделанный выше вывод (рис. 5). В серии расчетов объем использования угля изменялся в диапазоне от 111.29 до 134.50 млн т у. т., потребление газа – от 6.70 до 32.55 млн т у. т., потребление мазута – от 9.05 до 28.94 млн т у. т.

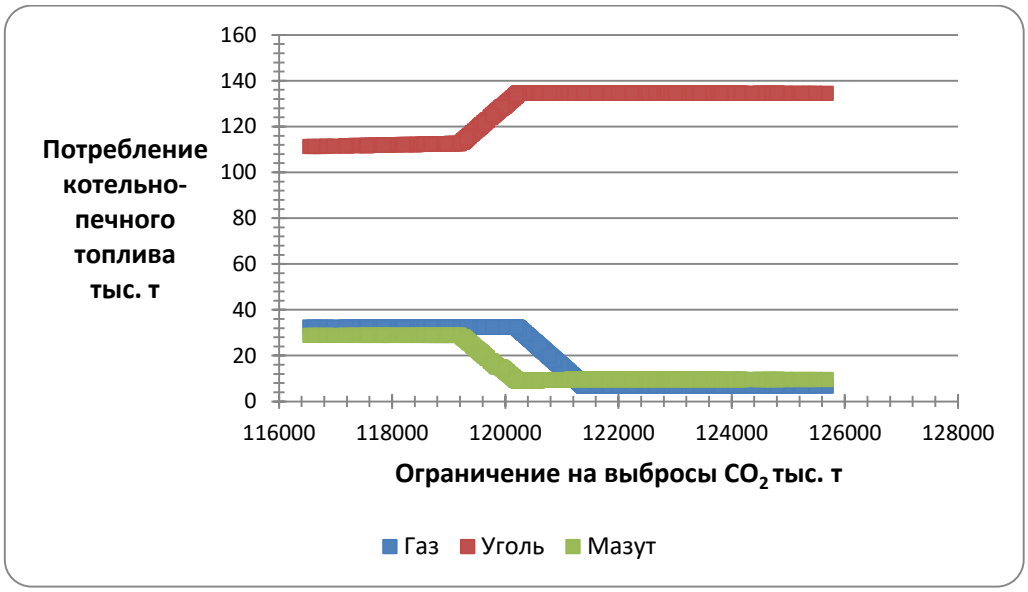


Рис. 5. Изменение структуры потребления котельно-печного топлива в зависимости от ограничения выбросов CO_2.

При более жестких ограничениях по выбросам наблюдаются снижение использования угля и увеличение роли газа и мазута, тогда как при менее жестких

значениях ограничений решение стремится к верхней границе использования угля и минимальным значениям альтернативных видов топлива. Следовательно, адаптация топливно-энергетического баланса к ужесточению экологического ограничения осуществляется главным образом за счет изменения структуры потребления топлива [11].

Таким образом, проведенный численный эксперимент показал, что предложенный численный метод позволяет исследовать не одно расчетное состояние, а ансамбль возможных состояний ТЭК при варьировании экологических ограничений. На примере Сибирского федерального округа показано, что ужесточение ограничения по выбросам CO_2 приводит прежде всего к замещению угля менее углеродоемкими энергоресурсами при сохранении устойчивости энергобаланса в большинстве сценарных реализаций. Это подтверждает применимость метода для анализа последствий экологических ограничений в условиях неопределенности.

ЗАКЛЮЧЕНИЕ

Решена задача интеграции семантического и математического моделирования в исследованиях направлений развития топливно-энергетического комплекса с учетом требований энергетической безопасности. Разработан программный компонент в составе комплекса ИНТЭК-SAW, обеспечивающий формализованное преобразование изменений когнитивной модели в параметры технико-экономической модели и обратную интерпретацию результатов расчетов.

Предложена технология проведения численного эксперимента, реализующая последовательный переход от построения онтологической и когнитивной моделей к формированию вычислительных сценариев, выполнению оптимизационных расчетов и представлению результатов. В отличие от существующих подходов, технология обеспечивает автоматизацию взаимодействия качественного и количественного анализа и поддерживает повторное использование знаний в составе единого программного средства.

Разработан численный метод стохастической корректировки параметров на основе когнитивных весов, позволяющий учитывать неопределенность экспертных оценок при формировании сценарных условий. Метод обеспечивает

переход от детерминированного задания воздействий к формированию ансамбля сценарных состояний ТЭК.

Проведенный численный эксперимент подтвердил работоспособность предложенного подхода и показал его применимость для анализа влияния экологических ограничений на структуру топливно-энергетического баланса. Установлено, что при ужесточении ограничения по выбросам CO_2 адаптация системы осуществляется преимущественно за счет изменения структуры потребления котельно-печного топлива при сохранении устойчивости энергобаланса в большинстве вариантов расчета.

Полученные результаты могут быть использованы при проведении численных экспериментов при изучении энергетической безопасности, а также при разработке инструментальных средств поддержки принятия решений в энергетике.

Благодарности

Работа выполнена в рамках проекта государственного задания (№ FWEU-2021-0007) программы фундаментальных исследований РФ на 2021–2030 гг.

СПИСОК ЛИТЕРАТУРЫ

1. *Массель Л.В.* Экосистема знаний для поддержки исследований и управления развитием энергосистем // XIV Всероссийское совещание по проблемам управления: материалы совещания. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2024. С. 3224–3231.

2. *Массель Л.В, Массель А.Г.* Семантические технологии на основе интеграции онтологического, когнитивного и событийного моделирования // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы III Междунар. науч.-техн. конф. (Минск, 21–23 февр. 2013 г.). Минск: БГУИР, 2013. С. 247–250.

3. *Ворожцова Т.Н., Пестерев Д.В., Кузьмин В.Р.* Семантическое моделирование в исследованиях устойчивости энергетических и социо-экологических систем // Информационные и математические технологии в науке и управлении. 2021. Т. 24, № 4. С. 31–43.

4. *Массель А.Г., Мамедов Т.Г.* Интеграция математического и когнитивного моделирования в исследованиях направлений развития ТЭК с позиции энергетической безопасности // Информационные и математические технологии в науке и управлении. 2025. Т. 39, № 3. С. 61–71.

5. *Кузьмин В.Р., Загорулько Ю.А.* Применение агентно-сервисного подхода при разработке интеллектуальных систем поддержки принятия решений в энергетике // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2020. Т. 18, № 3. С. 5–18.

6. *Tyurina E.A., Mednikov A.S., Elsukov P.Y., Sushko S.N.* Competitiveness of export-oriented systems of long-range energy supply // Energy Systems Research. 2024. Vol. 7, No. 1. P. 44–50.

7. *Макареня Т.А., Маннаа А.С., Калиниченко А.И., Петренко С.В.* Когнитивное моделирование социально-экономических систем: ретроспективный анализ инструментов и информационных систем // Вестник ВГУ. Серия: Системный анализ и информационные технологии. 2023. № 3. С. 84–94.

8. *Массель А.Г., Мамедов Т.Г., Пяткова Н.И.* Технология вычислительного эксперимента в исследованиях работы энергетических отраслей при реализации угроз энергетической безопасности // Информационные и математические технологии в науке и управлении. 2021. Т. 23, № 3. С. 62–73.

9. *Kosko B.* Fuzzy cognitive maps // International Journal of Man-Machine Studies. 1986. Vol. 24. P. 65–75.

10. *Zadeh L.A.* Fuzzy sets // Information and Control. 1965. Vol. 8. P. 338–353.

11. *Severina Y.D., Shakirov V.A., Takaishvili L.N.* Modeling the development of energy systems of remote areas in the context of the energy transition // Energy Systems Research. 2024. Vol. 7, No. 4. P. 5–12.

INTEGRATION OF SEMANTIC MATHEMATICAL MODELING FOR THE ANALYSIS OF ENERGY SECURITY PROBLEMS

A. G. Massel[1] [0009-0001-6884-1119], T. G. Mamedov[2] [0000-0001-9257-7410]

[1, 2]*Melentiev Energy Systems Institute of the Siberian Branch of the Russian Academy of Sciences, Irkutsk, Russia*

[1]amassel@gmail.com, [2]mamedowtymur@yandex.ru

Abstract

The study addresses the problem of integrating cognitive and mathematical modeling in research on the development directions of the fuel and energy complex, taking into account energy security requirements. The relevance of the work is due to the fact that in the existing two-level research methodology, the transition from the results of qualitative analysis using cognitive modeling to the parameters of the mathematical model is largely performed manually, which reduces the reproducibility of numerical experiments and limits the efficiency of accumulated knowledge usage. The aim of the work is to develop a software component that ensures the combined use of cognitive and mathematical models within an Energy Knowledge Ecosystem. A software component is proposed, implemented as part of the INTEC-SAW suite, which provides the transformation of changes in the cognitive model into the parameters of the economic-mathematical model, as well as the reverse interpretation of calculation results. Technology for conducting numerical experiments has been developed, including the construction of semantic (ontological and cognitive) models, formation of computational scenarios, execution of optimization calculations, and presentation of results, distinguished by the automation of the joint use of ontological, cognitive, and economic-mathematical models. To account for uncertainty, a numerical method of stochastic parameter adjustment based on cognitive weights is proposed. The effectiveness of the approach is demonstrated through a numerical experiment investigating the impact of CO_2 emission constraints on the energy balances of the Siberian Federal District. The practical significance of the work lies in increasing the validity and reproducibility of research on the development of the fuel and energy complex through the coordinated use of qualitative and quantitative analysis tools.

***Keywords**: energy complex, energy security, cognitive modeling, ontologies, computational experiment, linear programming*

REFERENCES

1. *Massel L.V.* Knowledge ecosystem for supporting research and management of energy systems development // Proceedings of the XIV All-Russian Conference on Control Problems. Moscow: V.A. Trapeznikov Institute of Control Sciences of RAS, 2024. P. 3224–3231.

2. *Massel L.V., Massel A.G.* Semantic technologies based on the integration of ontological, cognitive and event modeling // Open Semantic Technologies for Intelligent Systems (OSTIS-2013): Proceedings of the III International Scientific and Technical Conference (Minsk, February 21–23, 2013). Minsk: BSUIR, 2013. P. 247–250.

3. *Vorozhtsova T.N., Pesterev D.V., Kuzmin V.R.* Semantic modeling in studies of sustainability of energy and socio-ecological systems // Information and Mathematical Technologies in Science and Management. 2021. Vol. 24, No. 4. P. 31–43.

4. *Massel A.G., Mamedov T.G.* Integration of mathematical and cognitive modeling in studies of fuel and energy complex development from the perspective of energy security // Information and Mathematical Technologies in Science and Management. 2025. Vol. 39, No. 3. P. 61–71.

5. *Kuzmin V.R., Zagorulko Yu.A.* Application of the agent-service approach in the development of intelligent decision support systems in the energy sector // Vestnik of Novosibirsk State University. Series: Information Technologies. 2020. Vol. 18, No. 3. P. 5–18.

6. *Tyurina E.A., Mednikov A.S., Elsukov P.Y., Sushko S.N.* Competitiveness of export-oriented systems of long-range energy supply // Energy Systems Research. 2024. Vol. 7, No. 1. P. 44–50.

7. *Makarenya T.A., Mannaa A.S., Kalinichenko A.I., Petrenko S.V.* Cognitive modeling of socio-economic systems: retrospective analysis of tools and information systems // Vestnik VSU. Series: System Analysis and Information Technologies. 2023. No. 3. P. 84–94.

8. *Massel A.G., Mamedov T.G., Pyatkova N.I.* Technology of computational experiments in studies of energy sector functioning under energy security threats //

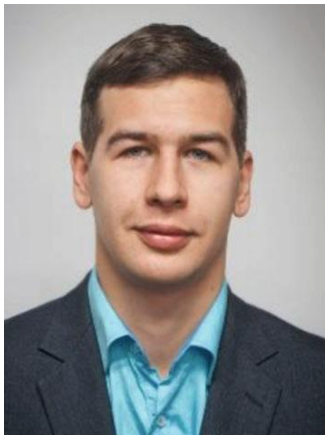
Information and Mathematical Technologies in Science and Management. 2021. Vol. 23, No. 3. P. 62–73.

9. *Kosko B.* Fuzzy cognitive maps // International Journal of Man-Machine Studies. 1986. Vol. 24. P. 65–75.

10. *Zadeh L.A.* Fuzzy sets // Information and Control. 1965. Vol. 8. P. 338–353.

11. *Severina Y.D., Shakirov V.A., Takaishvili L.N.* Modeling the development of energy systems of remote areas in the context of the energy transition // Energy Systems Research. 2024. Vol. 7, No. 4. P. 5–12.

СВЕДЕНИЯ ОБ АВТОРАХ



Массель Алексей Геннадьевич – окончил Иркутский государственный университет в 2007 году. Получил степень кандидата технических наук в 2011 году. Научные интересы включают семантическое моделирование, проектирование информационных систем и разработку систем поддержки принятия решений в энергетике. Автор более 70 работ. Scopus Author ID: 57205017287.

Alexey Genadevich MASSEL — born in 1985. Graduated from Irkutsk State University in 2007. Received the Ph.D. degree in Engineering in 2011. His research interests include semantic modeling, information system design, and the development of decision support systems in the energy sector. He is the author of more than 70 publications.

email: amassel@gmail.com

ORCID: 0000-0002-0351-0415

МАМЕДОВ Тимур Габилович – окончил Иркутский государственный технический университет в 2021 году. В настоящее время работает младшим научным сотрудником. Научные интересы включают семантическое моделирование, когнитивные карты и исследование направления развития ТЭК.

Timur Gabilovich MAMEDOV – graduated from Irkutsk National Research Technical University in 2021. He is currently working as a junior researcher. His research interests include semantic modeling, cognitive maps, and studies of fuel and energy complex (FEC) development.

Scopus Author ID: 58026460700.

email: mamedowtymur@yandex.ru

ORCID: 0000-0002-3396-5074

Материал поступил в редакцию 3 апреля 2026 года

УДК 004.81

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОИСКА ДЛЯ МАТЕМАТИЧЕСКОГО АРХИВА ПУБЛИКАЦИЙ

А. А. Насибулин[1] [0009-0005-9092-2520], О. М. Атаева[2] [0000-0003-0367-5575]

[1]*Московский физико-технический институт, г. Долгопрудный, Россия*

[2]*Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия*

[1]nasibulin.aa@phystech.edu, [2]oataeva@frccsc.ru

Аннотация

В работе проведено исследование, связанное с поиском схожих документов по математике. Разработан рекомендательный алгоритм нахождения похожих научных статей по данной тематике, использующий приоритетный поиск по математическим формулам с текстовым подкреплением.

Выполнен перевод текста из графического в текстовое представление через технологию OCR для последующего анализа и индексации. В процессе анализа реализовано разбиение текста на блоки с последующим извлечением из текста значимых формул, ключевых слов и фраз. В процессе индексации сформирована векторная база данных на основе векторных представлений формул, полученных через процесс эмбеддинга. Результаты индексации использованы при поиске статей, имеющих сходство с документом, подаваемым пользователем на вход алгоритма. Получен список похожих статей с сортировкой результатов по метрике близости векторных представлений формул.

Исходные данные представляют собой около 5000 научных статей, посвященных различным исследованиями по математической тематике и представленных в виде PDF-файлов.

Эксперимент проведен на основе данных конкретного контента библиотечной системы, но предложенная технология может быть распространена на другие библиотечные системы, в том числе содержащие статьи по другим тематикам, например, по физике и другим точным наукам.

Ключевые слова*: поиск по формулам, семантика, извлечение знаний, математический поиск, семантический поиск.*

ВВЕДЕНИЕ

В настоящее время остро стоит проблема поиска информации по научным статьям из-за роста объемов данных и стремительного увеличения сложности текстов в различных предметных областях, особенно в предметной области «математика». Классические алгоритмы лексического поиска по тексту (например, BM25), применяемые в большинстве рекомендательных систем, уступают алгоритмам семантического поиска с использованием Large Language Model (большая языковая модель, далее – LLM) [1]. Примером такого алгоритма является RAG с использованием LLM-модели Qwen3. Однако LLM-алгоритмы, как и классические, некорректно обрабатывают семантические значения математических формул, из-за чего в поисковой выдаче либо присутствуют некорректные совпадения формул по смыслу, либо отсутствуют похожие по смыслу формулы [2–4]. Отдельный поиск по формулам без использования текстового содержания хотя и эффективен для поиска по математическим статьям (например, Approach0 [12] достигает метрики качества $\mathrm{nDCG}' = 0.72$ на задаче поиска по формулам ARQMath-3), но все еще не задействует текстовые данные, использование которых может значительно улучшить поиск по статьям [5].

Таким образом, раздельное использование указанных выше алгоритмов поиска имеет ограничения, такие как потеря контекста из математических формул и текста или их некорректное распознавание. Поэтому для преодоления названных ограничений мы предлагаем комбинировать несколько из упомянутых подходов для достижения наилучших результатов поиска, т. е. создать систему гибридного поиска по текстам и формулам, которые в них содержатся. Таким образом, поставлена следующая задача: разработать и протестировать рекомендательный алгоритм нахождения похожих научных статей по математике, использующий приоритетный поиск по математическим формулам с текстовым подкреплением, т. е. реализовать поиск по математическим формулам, извлеченным из текста, с фильтрацией результатов этого поиска по семантическому сходству текстов. На вход алгоритма как для поиска, так и для индексирования подаются документы в распространенном формате – PDF. На выходе алгоритм выдает список найденных похожих статей, отсортированный по коэффициенту их совпадения с анализируемой статьей. Это и есть алгоритм поиска научных статей по математическим формулам с текстовым подкреплением.

БЛИЗКИЕ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

Для поиска по текстам, в том числе по научным текстам по математике, используют различные подходы поиска информации [5]. Наиболее распространенные варианты поиска информации можно разделить следующим образом: поиск по семантическим аннотациям из PDF-файлов [6], поиск с помощью RAG-систем [7] и поиск по онтологиям через граф знаний [8].

Системы Retrieval-Augmented Generation (генерация, дополненная поиском, далее – RAG) разделяются на два типа: классический RAG, в котором используется поиск по векторной базе данных (БД), и GraphRAG, использующий граф знаний. Оба типа систем на этапе анализа текстов при поиске или индексации не используют специфичную обработку формул, интерпретируя их как обычный текст, из-за этого поиск по формулам по существу не работает: например, после индексации более 5000 статей математического содержания системы не способны корректно различать формулы. Более того, из-за специфики хранения документов в RAG-системах поиск по точным совпадениям формул из проиндексированных документов невозможен.

Наиболее распространенные подходы такого поиска информации по онтологиям можно охарактеризовать следующим образом.

1. Поиск по графу знаний, создаваемому с использованием LLM [8–10]. Из-за того, что при построении графа знаний используются запросы к LLM, онтология, содержащаяся в ответе от нее, может быть некорректной (например, может быть извлечена только часть нужной формулы), так как у LLM-моделей есть склонность к ошибкам при ответе на математические вопросы [5].

2. Поиск по базе знаний, состоящей из текстов в векторном представлении, полученном через эмбеддинги [11]. Этот подход используется при построении RAG-систем и он не подразумевает специфичную обработку для формул, из-за чего на большом массиве данных поиск по формулам становится невозможным.

Что касается алгоритмов поиска по формулам, то их можно разделить на три типа по виду представления формул [5].

1. Представление математических формул в виде дерева расположения символов (Symbol Layout Tree, SLT) или дерева операторов (Operator Tree, OPT). Это наиболее распространенный вид представления формул. Примеры современных алгоритмов с его использованием: Approach0 [12], BERT (модель

MathBERT [13]), Tangent-CFT [14]. Эти алгоритмы предназначены только для поиска по формулам TeX-представлении, без контекста (текста), который может быть представлен вместе с формулой или вместо нее.

2. Представление в текстовом виде – используется, например, в алгоритме поиска по самой длинной общей подпоследовательности (Longest Common Subsequence, LCS) [15]. В современных алгоритмах такой вид представления не используется, так как поиск «по формулам как по текстам» неэффективен из-за особенностей их представления [16], которые невозможно учесть при использовании лексического поиска.

3. Представление в виде LEAN-кода (https://lean-lang.org/). Примером алгоритма, производящего поиск формул в таком представлении, является LeanSearch (существует еще версия без LLM) [17]. Этот вид представления используется в рекомендательных системах для ответов на математические вопросы, зачастую с использованием LLM. Применение такого представления в настоящей работе не рассматривается.

Существуют также системы гибридного поиска, из которых лучшим по совокупности показателей (SOTA – State-of-the-Art) решением для поиска по текстам с формулами является MABOWDOR [18]. Эта система использует комбинацию технологий: алгоритм поиска по формулам на основе SLT-дерева Approach0 [12], собственную BERT-подобную модель Coco-MAE [18] для поиска по формулам и алгоритм лексического поиска по текстам BM25+ [5] для поиска по текстам. Отметим, что эта система не адаптирована к текстам на русском языке.

Все приведенные примеры алгоритмов предполагают использование материалов в текстовом виде с формулами, представленными в TeX-нотации, и, соответственно, проводят оценку качества работы алгоритмов на таких инструментах оценки качества математического поиска, как ARQMATH [19] и NTCIR-12 [20]. Так как нами рассматриваются тексты научных работ по математике, которые содержат формулы, необходимо учесть, что значительная часть хранимых работ представлена в виде изображений или соответствующих PDF-файлов. Поэтому в таких случаях нужно использовать технологию распознавания текста из изображений/документов (OCR), чтобы гарантировать возможность работы разрабатываемого поискового алгоритма со статьями, представленными как в текстовом виде, так и в виде изображений, в том числе в PDF-файлах со специфичной ко-

дировкой или нестандартным форматированием. Так, на основании анализа результатов работы инструмента оценки качества работы алгоритмов OCR OmniDocBench [21] (точнее, интерпретации данных, собранных с помощью этой утилиты) нами была выбрана SOTA-технология набора инструментов распознавания текстов (OCR) PaddleOCR (PP-StructureV3). Эта технология производит экспорт формул в TeX-формат и сохраняет информацию о структуре документа, позволяя свести представление документов к единому формату, что необходимо для корректной работы системы поиска похожих работ.

Таким образом, использование комбинации распознавания PDF-файлов и гибридного поиска по математическим формулам с текстовым подкреплением ранее не рассматривалось (были использованы только TeX-файлы).

В настоящем исследовании для поиска по формулам использована эмбеддинг-модель MathBERT [13]. Она находит похожие формулы по семантике. Эта модель была обучена на учебниках и статьях по математике с сайта arxiv.org [13]. Ее особенность состоит в том, что она позволяет производить семантический поиск по формулам через их векторизацию и последующий поиск по расстоянию между ними (например, используя векторную базу данных scann [24]), как в RAG-системах. Пример работы поиска по формулам представлен на рис. 1. В этом примере был использован индекс из 100 различных формул.

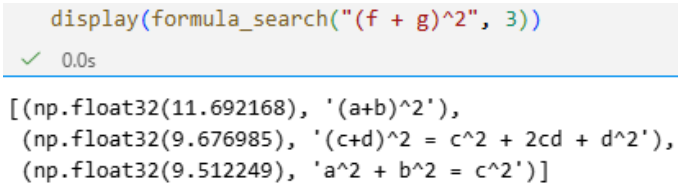


Рис. 1. Результат работы MathBERT.

Дополнительно была использована эмбеддинг-модель embeddinggemma-300m [22] для извлечения ключевых слов из текстов статей. Эта модель общего назначения, обученная для работы с мультиязычными текстами и имеющая лучшие показатели качества по сравнению с часто используемыми моделями bge-m3 и Qwen3Embedding 0.6B, при сниженном практически вдвое количестве параметров выдает при запросе более релевантные ключевые слова, как показано на рис. 2 (обеим моделям на вход подавалась аннотация статьи по семантическому поиску [23], дополнительный промпт не использован). Отметим, что

только модель embeddinggemma-300m экспортировала фразу «библиотека Libmeta». Экспорт этой фразы в данном примере примечателен тем, что ее наличие определяет возможность дальнейшего качественного поиска, так как она фактически является ключевой фразой для всего текста.

	model_name	keywords
0	sci_rus_small	междисциплинарного журнала, другие журналы, исследуется тематическое, журнала предлагается, журнала тематического, междисциплинарным исследованиям, статьи журнала, онтология журнала, экспертам журнала, интегрируются семантическую
1	frida_model	междисциплинарного журнала, журнала тематического, семантическую библиотеку, семантической библиотеке, онтология журнала, предлагается систематизация, междисциплинарной предметной, статьи журнала, тематической рубрики, анализа тематики
2	embeddinggemma_model	междисциплинарного журнала, журнала тематического, междисциплинарным исследованиям, анализа тематики, междисциплинарной предметной, журналов относящихся, анализа контента, контенту журнала, библиотеке libmeta многообразие междисциплинарного
3	bge_m3_model	междисциплинарного журнала, онтология журнала, междисциплинарным исследованиям, анализа тематики, исследуется тематическое, журнала тематического, тематического анализа, журнала интегрируются, журнала вырабатывается, междисциплинарной предметной
4	qwen3embedding_model	междисциплинарного журнала, знаний журнала, журнала тематического, тематического анализа, междисциплинарным исследованиям, анализа тематики, статьи журнала, тематической рубрики, онтология журнала, исследуется тематическое

Рис. 2. Сравнение работы BERT-моделей.

ОПИСАНИЕ ДАТАСЕТА

В качестве корпуса научных статей по математике был использован датасет научных работ на русском языке по математике и физике, содержащих математические формулы. Датасет представляет собой набор PDF-файлов порядка 5000 статей. Все статьи взяты из журнала «Известия высших учебных заведений. Математика» с 1997 по 2007 г. (1136 статей) и журнала «Вестник Тамбовского университета. Серия: Естественные и технические науки» с 2000 по 2013 г. (3892 статьи). Датасет содержит как вложенные в текст формулы (далее – inline-формулы), так и выделенные формулы (далее – outline-формулы).

МЕТОДИКА ПОИСКА

Гибридный поиск был реализован с помощью алгоритма поиска похожих математических статей, схема которого представлена на рис. 3.

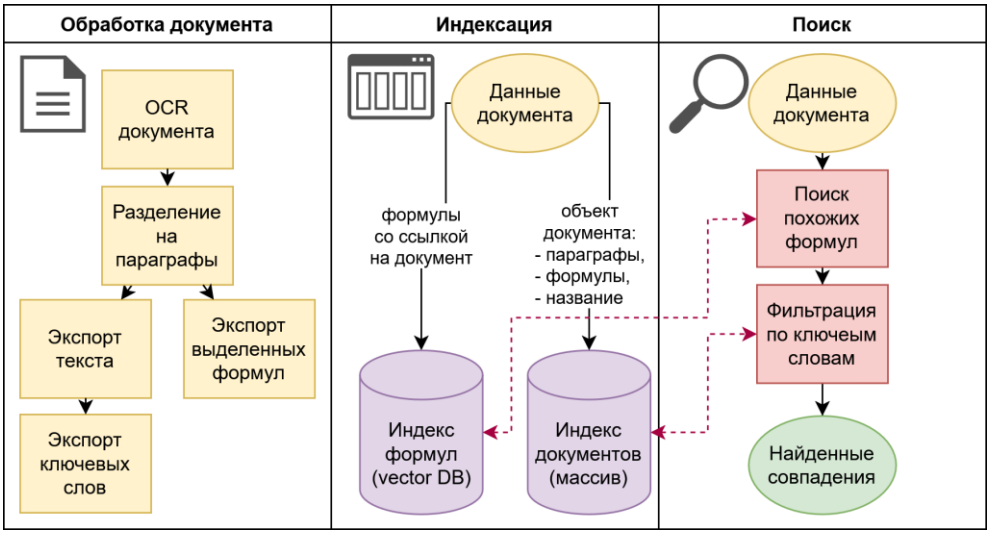


Рис. 3. Схема работы алгоритма

Документ можно разделить на несколько составляющих: текст (содержит inline-формулы), outline-формулы, таблицы, картинки (графики, схемы, фотографии и т. д.). Алгоритм в текущем виде работает только с текстом и формулами.

При обработке на вход алгоритму подается PDF-файл любого типа (страницы с произвольным текстовым и графическим содержанием, в том числе допустимо представление текстов в виде картинок или сканов). Из текста экспортируются и обрабатываются текст и формулы. Перечислим основные шаги работы алгоритма.

1. Обработка (анализ) документа. На этом этапе документ преобразуется в текстовое представление с помощью набора инструментов распознавания PaddleOCR, затем полученное текстовое представление конвертируется в объектное представление: текст документа разделяется на блоки (параграфы/значимые блоки текста и формулы), и из этих блоков с помощью BERT-модели EmbeddingGemma [22] производится экспорт ключевых слов. В сравнении с использованием всего текста как одного блока такой подход позволяет более точно извлекать контекст из каждого логически выделенного блока документа. Это достигается за счет уменьшенного количества текста на каждое вхождение при поиске. Пример обработки документа представлен на рис. 4.

2. Индексация. С помощью BERT-модели MathBERT [13] выполняется эмбеддинг найденных TeX-формул в векторные представления, которые добавляются в векторную базу данных scann [24] и в дальнейшем используются при выполнении поиска по документам. Каждая формула имеет ссылку на объект

блока документа, в котором она содержится. Используются только outline-формулы, т. е. формулы, выделенные из текста. Описанные данные представляют собой индекс математических формул. При проведении эксперимента индекс состоял из 144018 формул. Построение индекса заняло около 17 мин на процессоре AMD Ryzen 7840H.

3. Поиск. Сначала производится семантический поиск схожих формул по ранее полученному индексу математических формул. Результатом являются сопоставления «формула – блок статьи». Далее блоки статей фильтруются по ключевым словам, т. е. происходит лексический поиск по словам. Из отфильтрованного списка блоков извлекаются ссылки на статьи, в которых они содержатся. Результатом поиска является список похожих статей, отсортированных по величине метрики их сходства. Такой метрикой служит коэффициент расстояния между векторными представлениями формул (важно – это не значение расстояния между векторами, а коэффициент, обратно пропорциональный расстоянию между двумя векторами). Этот коэффициент вычисляется с использованием библиотеки базы данных scann.

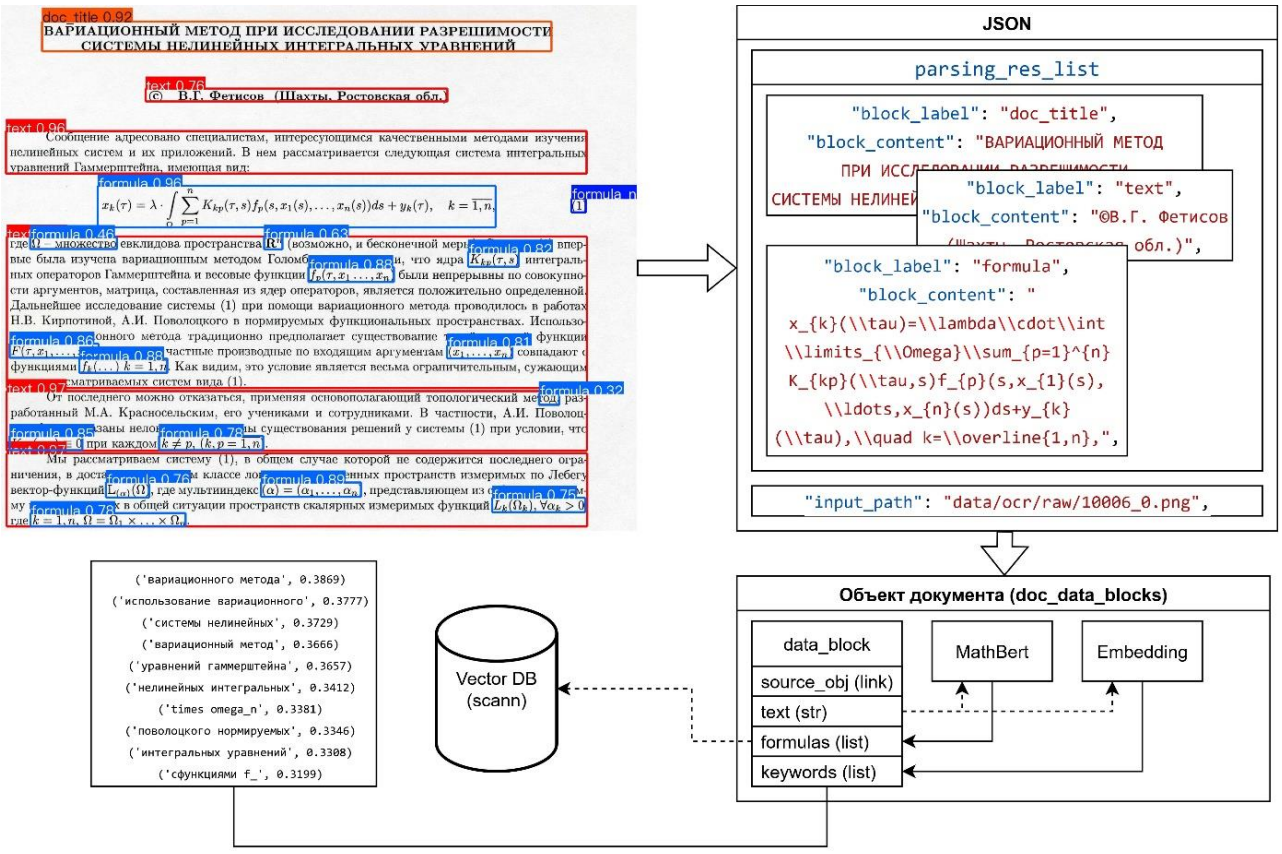


Рис. 4. Пример обработки документа.

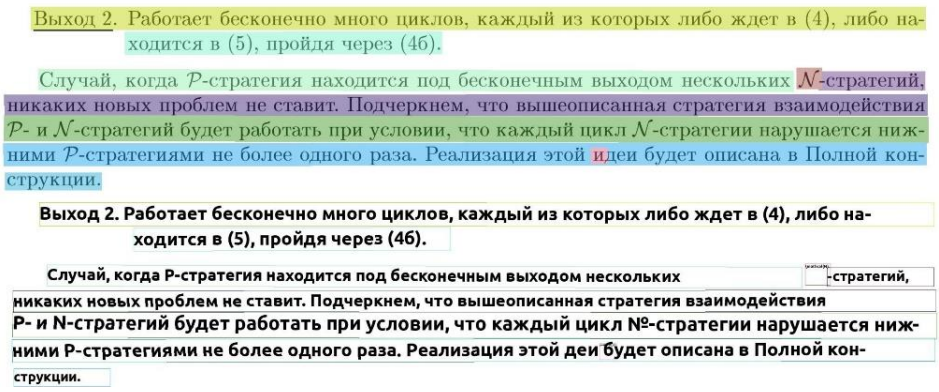
Далее отметим обнаруженные проблемы при разработке и тестировании алгоритма.

1. PaddleOCR имеет редкие ошибки распознавания содержащихся внутри текста inline-формул: часть из текста «переходит» в корпус формулы, а часть формул распознается как обычный текст, как показано на рис. 5. Для решения этой проблем требуется либо дообучение моделей распознавания, либо использование нормализации текстов. Для уменьшения последствий этой проблемы использовано не точное сравнение ключевых слов, а расстояние редактирования между ними [25] (см. пример на рис. 6).

2. Использование всего текста документа без дробления на части для извлечения ключевых слов через BERT-модель неэффективно, в том числе из-за ложных срабатываний (например, обнаруженное моделью сочетание слов «функция для» не может считаться ключевым) и потери большой части информации. Поэтому документы разделялись на блоки (главы) для сужения контекста, извлекаемого из текста, что позволило модели более точно выделять релевантные ключевые слова, за счет чего улучшается качество поиска.

3. Если в индексируемой статьей не было найдено outline-формул, то эта статья не появляется в списке найденных документов. Для обхода этой проблемы мы попытались использовать inline-формулы: при их добавлении количество найденных статей увеличивается, но при этом нерелевантные статьи начинают считаться релевантными, из-за этого был оставлен поиск только по outline-формулам.

Из датасета, содержащего более 5000 статей по математике на русском языке, экспертами были отобраны 50 пар статей, схожих по тематике. При тестировании на вход алгоритму подавался первый документ из пары для поиска похожих текстов. Если в топ-10 похожих документов содержался второй документ из пары, то поиск считался успешным. Таким образом, при тестировании алгоритма были успешно найдены 35 пар похожих математических статей. Для сравнения: при использовании лексического поиска (BM25+) пар не было найдено вообще, а при поиске по алгоритму RAG (doc2vec) было успешно найдено только 18 пар. Это подтверждает эффективность предложенного подхода к поиску похожих документов по сравнению с другими популярными алгоритмами.



Выход 2. Работает бесконечно много циклов, каждый из которых либо ждет в (4), либо находится в (5), пройдя через (4б).

Случай, когда $\mathcal{P}$-стратегия находится под бесконечным выходом нескольких $\mathcal{N}$-стратегий, никаких новых проблем не ставит. Подчеркнем, что вышеописанная стратегия взаимодействия $\mathcal{P}$- и $\mathcal{N}$-стратегий будет работать при условии, что каждый цикл $\mathcal{N}$-стратегии нарушается нижними $\mathcal{P}$-стратегиями не более одного раза. Реализация этой идеи будет описана в Полной конструкции.

Выход 2. Работает бесконечно много циклов, каждый из которых либо ждет в (4), либо находится в (5), пройдя через (4б).

Случай, когда P-стратегия находится под бесконечным выходом нескольких -стратегий, никаких новых проблем не ставит. Подчеркнем, что вышеописанная стратегия взаимодействия P- и N-стратегий будет работать при условии, что каждый цикл №-стратегии нарушается нижними P-стратегиями не более одного раза. Реализация этой деи будет описана в Полной конструкции.

Рис. 5. Примеры проблем при распознавании формул.

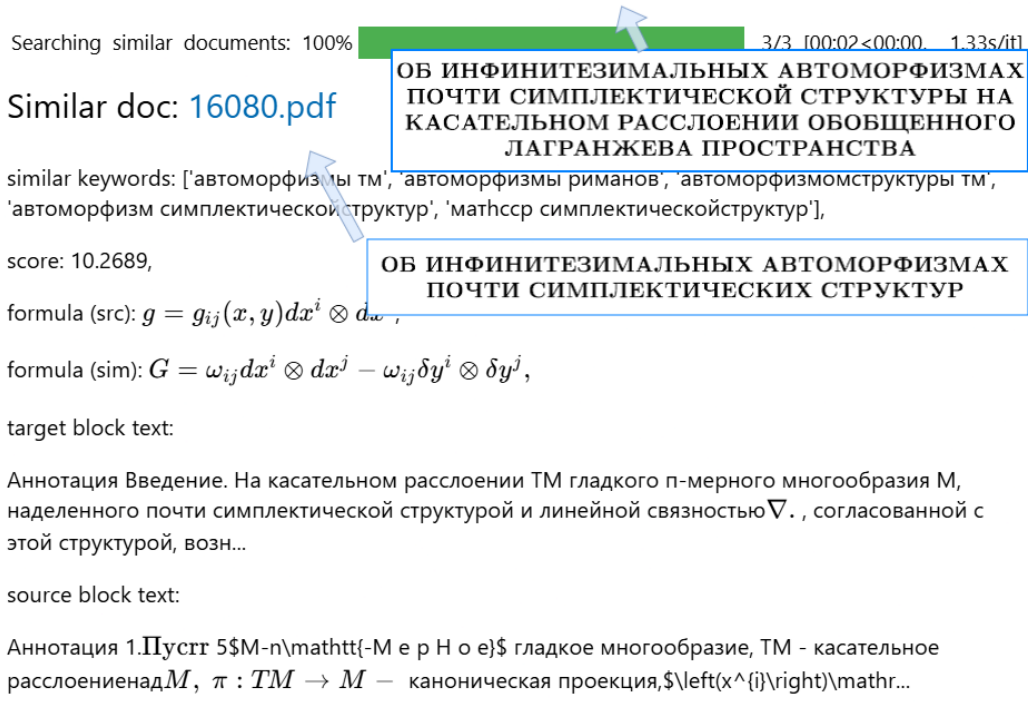
```
import difflib
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("russian")

words1, words2 = ["примеры", "тестим", "маJIинное", "обучение"], ["пример0в", "тест", "машина", "обучаться"]
stem_words1, stem_words2 = [stemmer.stem(word) for word in words1],  [stemmer.stem(word) for word in words2]
for word in stem_words1:
    print(f"Word: {word}, Closest: {difflib.get_close_matches(word, stem_words2, cutoff=0.7)}")
```

✓ 0.0s

```
Word: пример, Closest: ['пример0в']
Word: тест, Closest: ['тест']
Word: маJIин, Closest: ['машин']
Word: обучен, Closest: ['обуча']
```

Рис. 6. Примеры сравнения слов.

Пример результата работы алгоритма приведен на рис. 7.



```
Searching for similar documents of data/pdf_tex/raw/16824.pdf...
```

Searching similar documents: 100% 3/3 [00:02<00:00, 1.33s/it]

ОБ ИНФИНИТЕЗИМАЛЬНЫХ АВТОМОРФИЗМАХ ПОЧТИ СИМПЛЕКТИЧЕСКОЙ СТРУКТУРЫ НА КАСАТЕЛЬНОМ РАССЛОЕНИИ ОБОБЩЕННОГО ЛАГРАНЖЕВА ПРОСТРАНСТВА

Similar doc: 16080.pdf

similar keywords: ['автоморфизмы тм', 'автоморфизмы риманов', 'автоморфизмомструктуры тм', 'автоморфизм симплектическойструктур', 'mathccp симплектическойструктур'],

score: 10.2689,

ОБ ИНФИНИТЕЗИМАЛЬНЫХ АВТОМОРФИЗМАХ ПОЧТИ СИМПЛЕКТИЧЕСКИХ СТРУКТУР

formula (src): $g = g_{ij}(x, y)dx^i \otimes d$,

formula (sim): $G = \omega_{ij}dx^i \otimes dx^j - \omega_{ij}\delta y^i \otimes \delta y^j$,

target block text:

Аннотация Введение. На касательном расслоении TM гладкого n-мерного многообразия M, наделенного почти симплектической структурой и линейной связностью∇. , согласованной с этой структурой, возн...

source block text:

Аннотация 1.Пусгг 5$M-n\mathtt{-M е р Н о е}$ гладкое многообразие, TM - касательное расслоениенадM, $\pi : TM \to M$ — каноническая проекция,$\left(x^{i}\right)\mathr...

Рис. 7. Пример работы алгоритма.

ЗАКЛЮЧЕНИЕ

Основным полученным результатом является успешное нахождение похожих математических статей на тестируемом приватном датасете научных публикаций на русском языке с использованием алгоритма, основанного на поиске по формулам с текстовым подкреплением. Алгоритм решает проблему поиска похожих научных статей на русском языке, которые содержат математические формулы, и в отличие от его аналогов позволяет производить поиск по текстам, не переведенным в текстовый формат: он успешно находит похожие статьи математического характера, представленные в PDF-формате. Полученные результаты могут стать отправной точкой для дальнейших исследований по разработке алгоритма поиска похожих статей по математике в PDF-формате.

СПИСОК ЛИТЕРАТУРЫ

1. *Stuhlmann L., Saxer M. A., Fürst J.* Efficient and Reproducible Biomedical Question Answering using Retrieval Augmented Generation // arXiv:2505.07917v2. https://doi.org/10.48550/arXiv.2505.07917

2. *Polyanin A.D., Shingareva I.K.* The similarity index of mathematical and other scientific publications with equations and formulas and the problem of self-plagiarism identification // arXiv:2110.03872. https://doi.org/10.48550/ arXiv.2110.03872

3. *Wang R. et al.* Evaluation of LLMs for mathematical problem solving // arXiv:2506.00309. https://doi.org/10.48550/arXiv.2506.00309

4. *Forootani A.A*. survey on mathematical reasoning and optimization with Large Language Models // arXiv:2503.17726. https://doi.org/10.48550/arXiv.2503.17726

5. *Zanibbi R. et al*. Mathematical Information Retrieval: Search and Question Answering // arXiv:2408.11646v3. https://doi.org/10.48550/arXiv.2408.11646

6. *Невзорова О.А., Николаев К.С.* Семантическое аннотирование математических формул в PDF-документах // Электронные библиотеки. 2022. Т. 25, № 6. С. 616–639. https://doi.org/10.26907/1562-5419-2022-25-6-616-639

7. *Chen E. et al.* Comparing RAG and GraphRAG for Page-Level Retrieval Question Answering on Math Textbook // arXiv:2509.16780. https://doi.org/10.48550/arXiv.2509.16780

8. *Feng X. et al*. Ontology-grounded automatic Knowledge Graph construction by LLM under wikidata schema // arXiv:2412.20942. https://doi.org/10.48550/arXiv.2412.20942

9. *Lippolis A.S. et al*. Ontology Generation using Large Language Models // arXiv:2503.05388. https://doi.org/10.48550/arXiv.2503.05388

10. *Khasanshin A. et al.* Indexing mathematical scholarly papers as linked open data // Proceedings of the Sixth Russian Young Scientists Conference in Information Retrieval (VI Russian Summer School in Information Retrieval), 2012. P. 24–34. https://doi.org/10.18653/v1/P19-1023

11. *Trisedya B.D. et al.* Neural relation extraction for knowledge base enrichment, in: A. Korhonen, D. Traum, L. Màrquez (Eds.) // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, P. 229–240. https://doi.org/10.18653/v1/P19-1023

12. *Zhong W., Xie Y., Lin J. et al.* Applying Structural and Dense Semantic Matching for the ARQMath Lab 2022, CLEF // CLEF (Working Notes). 2022. P. 147-170.

13. *Shen J. T. et al*. MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education // arXiv:2106.07340. https://doi.org/10.48550/arXiv.2106.07340

14. *Mansouri B. et al.* Tangent-CFT: An embedding model for mathematical formulas //Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval. 2019. P. 11–18. https://doi.org/10.1145/3341981.3344235

15. *Kumar P., Agarwal A., Bhagvati C*. A structure based approach for mathematical expression retrieval // A Structure Based Approach for Mathematical Expression Retrieval // In: Sombattheera C., Loi N.K., Wankar R., Quan T. (Eds.) Multi-disciplinary Trends in Artificial Intelligence. MIWAI 2012. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2012. Vol. 7694. P. 23–34. https://doi.org/10.1007/978-3-642-35455-7_3

16. *Isele M.R.* Analyzing Similarity in Mathematical Content To Enhance the Detection of Academic Plagiarism // arXiv:1801.08439. https://doi.org/10.48550/arXiv.1801.08439

17. *Li I.R.* Towards Lightweight and LLM-Free Semantic Search for mathlib4 // AITP. 2025.

18. *Wei Zhong et al.* One Blade for One Purpose: Advancing Math Information Retrieval using Hybrid Search // In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23). 2023. P. 141–151. https://doi.org/10.1145/3539618.3591746

19. *Scharpf P. et al.* ARQMath Lab: An Incubator for Semantic Formula Search in zbMATH Open? // arXiv:2012.02413. https://doi.org/10.48550/arXiv.2012.02413

20. *Zanibbi R. et al.* NTCIR-12 MathIR Task Overview // NTCIR. 2016.

21. *Ouyang L. et al.* OmniDocBench: Benchmarking Diverse PDF Document Parsing with Comprehensive Annotations // arXiv:2412.07626. https://doi.org/10.48550/arXiv.2412.07626

22. *Vera H.S. et al.* EmbeddingGemma: Powerful and Lightweight Text Representations // arXiv:2509.20354. https://doi.org/10.48550/arXiv.2509.20354

23. *Ataeva O.M. et al.* Data mining when constructing a knowledge graph of a multidisciplinary journal // Information and mathematical technologies in science and management. 2024. Vol. 3 (35). P. 5–19.

24. *Refahi S.M. et al.* Fast and Scalable Gene Embedding Search: A Comparative Study of FAISS // arXiv:2507.16978. https://doi.org/10.48550/arXiv.2507.16978

25. *Python developers.* Documentation of library difflib // Python 3.14.3 documentation.

DEVELOPMENT OF AN INTELLIGENT SEARCH SYSTEM FOR THE MATHEMATICAL ARCHIVE OF PUBLICATIONS

A. A. Nasibulin[1][0009-0005-9092-2520], O. M. Ataeva[2][0000-0003-0367-5575],

[1]*Moscow Institute of Physics and Technology, Dolgoprudny, Russia*

[2]*Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia*

[1]nasibulin.aa@phystech.edu, [2]oataeva@frccsc.ru

Abstract

A study was conducted on searching for similar documents. The goal was to create a recommendation algorithm for finding similar scientific articles in mathematics using a prioritized search of mathematical formulas with textual support.

The text was converted from graphical to textual representation using OCR technology for subsequent analysis and indexing. During the analysis process, the text was divided into blocks, followed by the extraction of significant formulas, keywords, and phrases from the text. During the indexing process, a vector database was formed based on vector representations of formulas obtained through the embedding process. The indexing results were used to search for articles that are similar to the document submitted by the user to the algorithm input. A list of similar articles is displayed with results sorted by the metric of closeness of vector representations of formulas.

The source data consisted of approximately 5,000 scientific articles devoted to various studies on mathematical topics and presented as PDF files. The experiment was conducted based on data from specific library system content, but the proposed technology can be extended to other library systems, including those containing articles on other topics, such as physics and other exact sciences.

***Keywords**: formula search, semantics, knowledge extraction, mathematical search, semantic search.*

REFERENCES

1. *Stuhlmann L., Saxer M.A., Fürst J.* Efficient and Reproducible Biomedical Question Answering using Retrieval Augmented Generation // arXiv:2505.07917v2. https://doi.org/10.48550/arXiv.2505.07917

2. *Polyanin A.D., Shingareva I.K.* The similarity index of mathematical and other scientific publications with equations and formulas and the problem of self-plagiarism identification // arXiv:2110.03872.
https://doi.org/10.48550/ arXiv.2110.03872

3. *Wang R. et al.* Evaluation of LLMs for mathematical problem solving // arXiv:2506.00309. https://doi.org/10.48550/arXiv.2506.00309

4. *Forootani A.A*. Survey on mathematical reasoning and optimization with Large Language Models // arXiv:2503.17726.
https://doi.org/10.48550/arXiv.2503.17726

5. *Zanibbi R. et al*. Mathematical Information Retrieval: Search and Question Answering // arXiv:2408.11646v3. https://doi.org/10.48550/arXiv.2408.11646

6. *Nevzorova O.A., Nikolaev K.S.* Semantic Annotation of Mathematical Formulas in PDF-Documents // Russian Digital Libraries. 2022. Vol. 25. No. 6. P. 616–639.

https://doi.org/10.26907/1562-5419-2022-25-6-616-639

7. *Chen E. et al.* Comparing RAG and GraphRAG for Page-Level Retrieval Question Answering on Math Textbook // arXiv:2509.16780. https://doi.org/10.48550/arXiv.2509.16780

8. *Feng X. et al*. Ontology-grounded automatic Knowledge Graph construction by LLM under wikidata schema // arXiv:2412.20942. https://doi.org/10.48550/arXiv.2412.20942

9. *Lippolis A.S. et al*. Ontology Generation using Large Language Models // arXiv:2503.05388. https://doi.org/10.48550/arXiv.2503.05388

10. *Khasanshin A. et al.* Indexing mathematical scholarly papers as linked open data // Proceedings of the Sixth Russian Young Scientists Conference in Information Retrieval (VI Russian Summer School in Information Retrieval), 2012. P. 24–34. https://doi.org/10.18653/v1/P19-1023

11. *Trisedya B.D. et al.* Neural relation extraction for knowledge base enrichment, in: A. Korhonen, D. Traum, L. Màrquez (Eds.) // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, P. 229–240. https://doi.org/10.18653/v1/P19-1023

12. *Zhong W., Xie Y., Lin J. et al.* Applying Structural and Dense Semantic Matching for the ARQMath Lab 2022, CLEF // CLEF (Working Notes). 2022. P. 147-170.

13. *Shen J.T. et al*. MathBERT: A Pre-trained Language Model for General NLP Tasks in Mathematics Education // arXiv:2106.07340. https://doi.org/10.48550/arXiv.2106.07340

14. *Mansouri B. et al.* Tangent-CFT: An embedding model for mathematical formulas //Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval. 2019. P. 11–18. https://doi.org/10.1145/3341981.3344235

15. *Kumar P., Agarwal A., Bhagvati C*. A structure based approach for mathematical expression retrieval // A Structure Based Approach for Mathematical Expression Retrieval // In: Sombattheera C., Loi N.K., Wankar R., Quan T. (Eds.) Multi-disciplinary Trends in Artificial Intelligence. MIWAI 2012. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2012. Vol. 7694. P. 23–34. https://doi.org/10.1007/978-3-642-35455-7_3

16. *Isele M.R.* Analyzing Similarity in Mathematical Content to Enhance the Detection of Academic Plagiarism // arXiv:1801.08439.

https://doi.org/10.48550/arXiv.1801.08439

17. *Li I.R.* Towards Lightweight and LLM-Free Semantic Search for mathlib4 // AITP. 2025.

18. *Wei Zhong et al.* One Blade for One Purpose: Advancing Math Information Retrieval using Hybrid Search // In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23). 2023. P. 141–151. https://doi.org/10.1145/3539618.3591746

19. *Scharpf P. et al.* ARQMath Lab: An Incubator for Semantic Formula Search in zbMATH Open? // arXiv:2012.02413. https://doi.org/10.48550/arXiv.2012.02413

20. *Zanibbi R. et al.* NTCIR-12 MathIR Task Overview // NTCIR. 2016.

21. *Ouyang L. et al.* OmniDocBench: Benchmarking Diverse PDF Document Parsing with Comprehensive Annotations // arXiv:2412.07626. https://doi.org/10.48550/arXiv.2412.07626

22. *Vera H.S. et al.* EmbeddingGemma: Powerful and Lightweight Text Representations // arXiv:2509.20354. https://doi.org/10.48550/arXiv.2509.20354

23. *Ataeva O.M. et al.* Data mining when constructing a knowledge graph of a multidisciplinary journal // Information and mathematical technologies in science and management. 2024. Vol. 3 (35). P. 5–19.

24. *Refahi S.M. et al.* Fast and Scalable Gene Embedding Search: A Comparative Study of FAISS // arXiv:2507.16978. https://doi.org/10.48550/arXiv.2507.16978

25. Python developers. Documentation of library difflib // Python 3.14.3 documentation.

СВЕДЕНИЯ ОБ АВТОРАХ

НАСИБУЛИН Алексей Алексеевич – студент 2 курса магистратуры Московского физико-технического института по направлению «Науки о данных». Основные направления научных исследований: обработка естественного языка, компьютерное зрение, искусственный интеллект.

Aleksey Alekseevich NASIBULIN – second-year master's studentat MIPT in the field of «Data Science». Major fields of scientific research are Natural Language processing, computer vision and artificial intelligence.

email: nasibulin.aa@phystech.edu

ORCID: 0009-0005-9092-2520

АТАЕВА Ольга Муратовна – старший научный сотрудник Вычислительного центра им. А.А. Дородницына ФИЦ ИУ РАН, к. т. н. Область научных интересов: системное программирование, базы данных, инженерия знаний и онтологии.

Olga Muratovna ATAEVA – senior researcher at the Dorodnicyn Computing Center, Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences; Candidate of Technical Sciences (PhD). Research interests: systems programming, databases, knowledge engineering, ontologies. Number of scientific publications – 80.

email: oataeva@frccsc.ru

ORCID: 0000-0003-0367-5575

Материал поступил в редакцию 18 марта 2026 года

УДК 004.4

МОДЕЛЬ И АРХИТЕКТУРА МНОГОУРОВНЕВОГО АНАЛИЗА СХОДСТВА ANDROID-ПРИЛОЖЕНИЙ ПО СТАТИЧЕСКИМ ПРИЗНАКАМ

В. В. Петров[0009-0004-4213-7328]

Казанский (Приволжский) федеральный университет, г. Казань, Россия

valeryvpetrov.itis@gmail.com

Аннотация

Рассмотрена задача многоуровневого анализа сходства приложений для платформы Android по статическим признакам в цифровых коллекциях мобильных приложений. В таких коллекциях встречаются дубликаты, ответвленные версии, перепакованные приложения и иные модифицированные варианты; вредоносная нагрузка рассматривается как возможный частный случай модификации, а не как синоним перепаковки.

Формализована функция сходства приложений по статическим признакам, построена статическая модель приложения и предложена архитектура анализа, разделяющая предварительный отбор кандидатов, углубленное сопоставление, интерпретацию результата и слой формирования заключения. Показано, что значимая информация о близости приложений содержится не только в байткоде classes.dex, но и в манифесте AndroidManifest.xml, ресурсах, APK-внутренних метаданных и библиотечных зависимостях. Численная оценка сходства вычисляется только при успешном построении статических моделей сравниваемых приложений; в противном случае фиксируется отдельный служебный технический статус с нормализованной причиной отказа.

На локальном пилотном наборе из пяти основных пар и двух граничных случаев наблюдалось, что явный учет библиотечных зависимостей и раздельная фиксация технических ограничений прототипа позволяют получить более интерпретируемый результат, однако эти данные следует рассматривать как предварительные и не дающие оснований для окончательной валидации архитектуры на больших коллекциях.

Ключевые слова: *приложение для платформы Android, статический анализ, анализ сходства программ, поиск модифицированных вариантов, перепакованные приложения, библиотечная зависимость, интерпретация результата, цифровая коллекция приложений*.

ВВЕДЕНИЕ

Задача поиска модифицированных вариантов приложений для платформы Android представляет практический и исследовательский интерес при работе с цифровыми коллекциями мобильных приложений. К таким вариантам относятся:

- дубликаты, т. е. копии без существенных содержательных изменений;
- ответвленные версии, сохраняющие общее происхождение, но развивающиеся по самостоятельной линии;
- перепакованные приложения, в которых исходный программный пакет модифицируется путем добавления, удаления или замены отдельных компонентов;
- иные модифицированные варианты, для которых вредоносная нагрузка является возможным частным случаем модификации, а не обязательным признаком перепаковки.

Для практики разработки, сопровождения, контроля качества и анализа безопасности требуется не только устанавливать факт сходства таких приложений, но и выявлять, какие именно признаки обусловливают их близость или различие. В сценариях перепаковки и привнесения постороннего кода к легитимному приложению добавляется сторонняя функциональность, изменяются ресурсы, расширяется набор разрешений или подключаются новые библиотеки [1]. В этих случаях аналитически значимы не только ответ на вопрос о наличии сходства, но и структура интерпретации результата: связано ли сходство с собственным кодом приложения, библиотечными включениями, изменениями манифеста либо с ресурсным слоем. Для тематики электронных библиотек это соответствует сценарию, когда в цифровом фонде APK-артефактов (Android Package Kit Artifact) требуется найти близкие версии одного и того же приложения, выделить перепакованные экземпляры и объяснить,

какие именно статические признаки послужили основанием для такого сопоставления.

Для тематики электронных библиотек это соответствует сценарию, когда в цифровом фонде APK-артефактов требуется найти близкие версии одного и того же приложения, выделить перепакованные экземпляры и объяснить, какие именно статические признаки послужили основанием для такого сопоставления. В этом случае нужны не только ранжированный поиск по коллекции, но и воспроизводимая интерпретация причин сходства.

Во многих прикладных системах поиска модифицированных вариантов анализ сводится к вычислению одной численной оценки сходства для пары приложений. Такая схема обладает ограниченной пригодностью для реального анализа. Во-первых, при росте коллекции приложений полный попарный перебор быстро становится вычислительно дорогим. Во-вторых, единый итоговый показатель скрывает структуру результата: аналитик не видит, связано ли сходство с собственным кодом приложения, повторным использованием библиотек, со сходством манифеста или ресурсным слоем. В-третьих, при неуспешном построении статической модели приложения технический исход может быть ошибочно интерпретирован как содержательно низкое сходство.

Переход к текущей постановке опирается на ранее опубликованные работы автора. В работе [2] был анонсирован базовый режим численной оценки сходства приложений для платформы Android. В [3, 4] этот подход был развит до метода, основанного на анализе байткода и сравнении графов потока управления. Эти работы служат воспроизводимой отправной точкой байткод-ориентированного этапа исследования сходства Android-приложений. Однако текущий этап разработки метода показывает, что их недостаточно рассматривать как окончательную форму анализа сходства. Приложение для платформы Android представляет собой более сложный объект, чем только classes.dex: в практически значимых сценариях сигналы сходства содержатся также в AndroidManifest.xml, ресурсах, APK-внутренних метаданных и библиотечных зависимостях.

В связи с этим актуальной становится задача перехода от сравнения, опирающегося только на байткод, к многоуровневому анализу сходства по статическим признакам.

В настоящей статье рассмотрены формализация этой задачи, архитектура соответствующего анализа и требования к экспериментальному контуру, который не смешивает предварительный отбор кандидатов, углубленное сопоставление, интерпретацию результата и служебные технические статусы вычислительного контура.

К числу основных результатов работы относятся следующие положения. Во-первых, формализована функция сходства приложений по статическим признакам. Во-вторых, введена статическая модель приложения как рабочий объект многоуровневого анализа. В-третьих, предложена архитектура многоуровневого контура, в которой предварительный отбор, углубленное сопоставление, интерпретация результата и слой формирования заключения рассматриваются как различные уровни обработки. В-четвертых, уточнены требования к экспериментальному контуру проверки такой архитектуры и приведены предварительные результаты пилотного экспериментального цикла без обобщающих утверждений о завершенной валидации метода.

По сравнению с работами [2–4] в настоящей работе добавляется три конкретных элемента. Во-первых, байткод-ориентированное сравнение расширено до многоуровневой статической модели, включающей манифест, ресурсы, APK-внутренние метаданные и библиотечные зависимости. Во-вторых, архитектура анализа явно разделяет предварительный отбор кандидатов и углубленное сопоставление, что позволяет рассматривать задачу не только как попарное сравнение, но и как поиск по доверенной коллекции. В-третьих, в экспериментальный контур введена отдельная фиксация служебных технических статусов и граничных случаев, что позволяет не смешивать ограничения текущего прототипа с содержательным результатом сравнения.

МЕТОДОЛОГИЧЕСКИЕ ОСНОВАНИЯ

Общую теоретическую рамку для анализа сходства программ задают работы по сходству программ и программным отпечаткам, где сходство рассматривается как результат извлечения признаков и сравнения представлений программных объектов [5]. Для приложений платформы Android эта линия получила развитие в исследованиях, ориентированных на статический

анализ байткода, перепаковку приложений и обнаружение вредоносных модификаций [1, 6, 7].

С одной стороны, имеются работы, в которых сходство выводится из сравнения программных структур и кода. К ней относятся, в частности, подходы на основе статического анализа байткода, сравнения сигнатур методов и графов потока управления [4, 6]. Эти методы ценны тем, что опираются на содержательные программные признаки и позволяют строить численную оценку сходства. Однако для практически значимых сценариев они часто оказываются либо вычислительно дорогими, либо недостаточно устойчивыми к обфускации, структурным преобразованиям и влиянию общих библиотечных компонентов.

С другой стороны, развиваются подходы, ориентированные на предварительный отбор кандидатов или использование альтернативных статических описаний приложения. Так, в работе [8] сравнение приложений построено по ресурсному слою, что показывает самостоятельную диагностическую ценность не только кода, но и ресурсов приложения. Для прикладных сценариев проверки коллекций приложений для платформы Android развивались и более масштабируемые линии анализа, где предварительный отбор близких приложений выполняется отдельно от последующего углубленного разбора (см., например, [9]).

Значимый исследовательский шаг связан с переходом от одного итогового показателя сходства к интерпретируемому анализу. В работе [10] показано, что практически полезный инструмент должен не только возвращать итоговое значение сходства, но и указывать структуру различий: совпадающие методы, новые методы, удаленные методы и другие типы изменений. Эта линия имеет особое значение для задач экспертной проверки, где требуется не только ранжирование кандидатов, но и интерпретация причин их близости.

Отдельный класс работ посвящен перепаковке и привнесению постороннего кода в приложения платформы Android. Для этих сценариев принципиальным является то, что сходство может проявляться не только на уровне кода, но и на уровне ресурсов, компонентов манифеста, разрешений и библиотечных включений [1]. Тем самым исследования в этой области усиливают аргумент против узкой постановки, опирающейся только на

байткод, и поддерживают переход к многослойной статической модели приложения.

Еще одна значимая линия связана с обработкой влияния общих библиотечных компонентов. Под этим влиянием далее понимается ситуация, в которой совпадающий библиотечный код искусственно увеличивает итоговую оценку сходства или скрывает различия между собственным кодом приложений. Работы по детекции библиотек в Android-приложениях показывают, что сторонние библиотечные компоненты широко распространены, существенно влияют на результаты статического анализа и требуют отдельного выявления для задач безопасности, обнаружения перепаковки и последующего сравнения приложений [11–13]. Следовательно, модуль учета библиотек не должен быть скрытой эвристикой внутри итогового показателя; его необходимо рассматривать как отдельный воспроизводимый компонент анализа.

Наконец, обзор литературы выявил и проблему сопоставимости результатов. Несмотря на большое число публикаций по обнаружению перепакованных приложений, результаты многих исследований плохо сопоставимы из-за закрытых наборов данных, слабой воспроизводимости и различия в уровне детализации эталонных разметок [9]. Это означает, что новая система анализа сходства должна проектироваться одновременно с явным контуром экспериментальной проверки, который различает основной набор валидных пар и набор граничных случаев.

Таким образом, отмеченные выше исследования поддерживают несколько принципиальных выводов. Во-первых, сходство не должно сводиться к одному числу без интерпретации структуры результата. Во-вторых, практически применимая система должна разделять предварительный и углубленный анализ. В-третьих, приложение необходимо описывать не только через код, но и через иные статические признаки. В-четвертых, влияние общих библиотечных компонентов и служебные технические статусы вычислительного контура должны учитываться явно, а не скрываться внутри единой метрики.

ФОРМАЛИЗАЦИЯ ЗАДАЧИ

Пусть Ω – множество приложений для платформы Android, рассматриваемых в задаче статического анализа. Требуется задать функцию сходства

$$S: \Omega \times \Omega \to [0, 1],$$

где $S(A, B)$ определяет нормированную оценку сходства приложений A и B по выбранному набору статических признаков.

В этой постановке равенство $S(A, B) = 1$ означает идентичность приложений A и B в рамках рассматриваемых статических характеристик, а $S(A, B) = 0$ – отсутствие общих значимых статических характеристик в рамках принятой формализации. Эти значения не следует трактовать как абсолютную идентичность или полное отсутствие общих элементов вне границ рассматриваемой модели сходства.

Для реализации такой функции введем оператор M построения статической модели приложения. Для каждого приложения $X \in \Omega$ оператор M либо строит статическую модель $M(X)$, либо в текущем прототипе возвращает служебный технический статус FAIL, означающий невозможность корректно построить модель средствами текущего вычислительного контура. Статическая модель приложения в настоящей работе понимается как структурированное описание приложения, включающее выбранные статические признаки и отношения между ними. Такая модель может строиться по одному слою или согласованной комбинации нескольких слоев: байткоду, компонентам и свойствам AndroidManifest.xml, ресурсам, APK-внутренним метаданным, библиотечным зависимостям или по их сочетанию. Внешние магазинные атрибуты и внешние пользовательские метки в данную модель не входят.

В настоящей работе численная оценка сходства вычисляется только тогда, когда модели для обоих приложений построены успешно. В этом случае используется оператор сравнения Φ и выполняется соотношение

$$S(A, B) = \Phi(M(A), M(B)), \qquad 0 \le \Phi(M(A), M(B)) \le 1.$$

Если оператор M не может корректно построить модель хотя бы для одного приложения пары, значение $S(A, B)$ не вычисляется. Вместо него

фиксируются служебный технический статус FAIL и нормализованная причина отказа. В текущем прототипе различаются по меньшей мере следующие причины: input_model_failed для входного приложения, fast_signature_failed для неуспешного построения быстрой сигнатуры и candidate_model_missing для отсутствия корректной модели кандидата в репозитории доверенной коллекции. Это различение необходимо: отсутствие сходства и невозможность корректно выполнить анализ относятся к разным типам исходов работы системы. Первый исход является результатом сравнения приложений, второй — техническим отказом вычислительного контура. Поэтому они должны фиксироваться разными статусами и интерпретироваться отдельно.

Исследовательские задачи в такой постановке состоят в следующем:

- определить, какие статические модели приложения являются содержательно значимыми для анализа сходства;
- построить нормированную оценку сходства приложений по их статическим признакам;
- развести содержательные исходы анализа и служебные технические статусы текущего вычислительного контура;
- задать воспроизводимый экспериментальный контур для проверки предварительного отбора, углубленного сопоставления и качества интерпретации.

Функции практически применимой системы анализа при этом включают:

- предварительный отбор кандидатов для сокращения пространства сравнения;
- углубленное сопоставление ограниченного набора наиболее релевантных пар;
- интерпретацию результата в терминах использованных моделей и влияния общих библиотечных компонентов;
- формирование заключения для эксперта на основе оценки сходства и структурированной интерпретации.

Тем самым система анализа сходства рассматривается не как единичный вычислитель итогового показателя, а как многоуровневый контур обработки. Значимым следствием этой постановки является отказ от предположения, что один и тот же слой признаков должен одновременно обеспечивать и масштабируемый поиск по большой коллекции, и глубокий содержательный разбор каждой пары.

АРХИТЕКТУРА И АЛГОРИТМ

Предлагаемая архитектура базируется на том, что практически применимая система анализа сходства должна быть составной и многоуровневой. Сначала целесообразно показать сам ход анализа входного приложения относительно доверенной коллекции. Именно эту процессную сторону архитектуры отражает рис. 1.

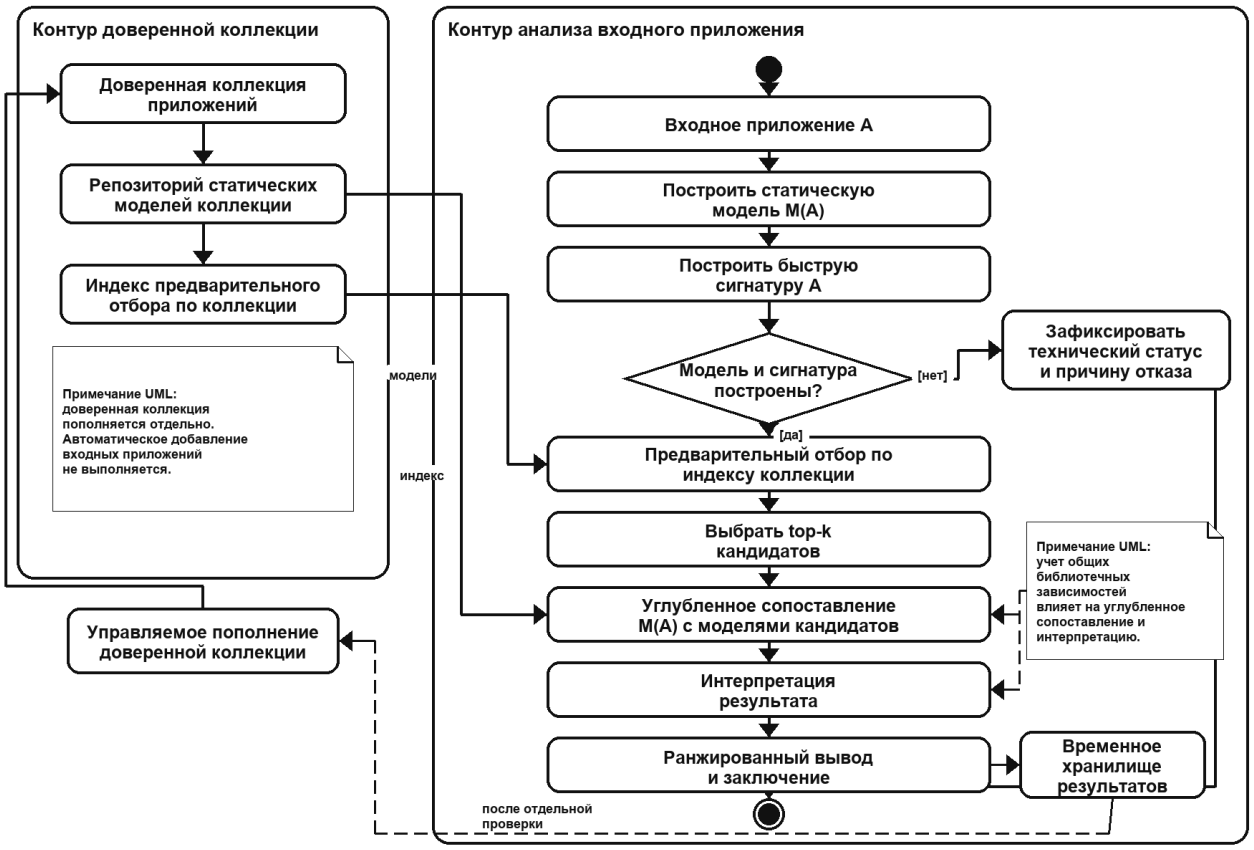


Рис. 1. UML-диаграмма деятельности двухэтапного анализа входного приложения относительно доверенной коллекции приложений для платформы Android.

На рис. 1 представлены два связанных контура: контур доверенной коллекции и контур анализа входного приложения. В первом контуре хранятся ранее построенные статические модели коллекции и индекс предварительного отбора, обеспечивающие накопление знаний системы. Во втором контуре для входного приложения *A* строятся статическая модель и быстрая сигнатура, после этого по индексу коллекции отбираются кандидаты для углубленного сопоставления. Таким образом диаграмма деятельности описывает не полный попарный перебор по коллекции, а управляемый переход от быстрого отбора к детальному сопоставлению ограниченного числа кандидатов. Отдельная техническая ветвь в этой схеме отражает служебный статус текущего вычислительного контура, а не самостоятельный содержательный результат сравнения. Входное приложение не включается автоматически в доверенную коллекцию: допускаются лишь временное хранение результатов и отдельное управляемое пополнение основной базы. Из этой процессной схемы следуют основные архитектурные принципы системы многоуровневого анализа сходства.

1. Многоуровневая статическая модель приложения: приложение для платформы Android не должно моделироваться только через код classes.dex. Для задач клонирования, перепаковки и анализа вредоносных модификаций значимы также компоненты и атрибуты AndroidManifest.xml, статические ресурсы, метаданные сборки и библиотечные зависимости.

2. Явное разделение предварительного и углубленного анализа: полный углубленный разбор всех пар приложений в коллекции плохо масштабируется, поэтому предварительный слой должен отбирать кандидатов по сравнительно дешевым признакам, а углубленный слой — запускаться только на ограниченном наборе пар.

3. Обязательная интерпретация результата: итог анализа не должен ограничиваться единственным числом. Необходимо возвращать и нормированную оценку сходства, и структурированную интерпретацию, показывающую, в каких слоях приложения обнаружено сходство или различие.

4. Явный учет влияния общих библиотечных компонентов: повторное использование библиотек может искусственно завышать оценку сходства между независимыми приложениями и скрывать различия между основной логикой

приложения и внешними зависимостями. Поэтому влияние библиотечного слоя должно быть трассируемым и интерпретируемым.

5. Разведение содержательных исходов сравнения и технических ограничений вычислительного контура: система должна различать собственно результат сравнения и служебный статус, возникающий при невозможности корректно построить модель или извлечь признаки средствами текущего прототипа.

6. Хранение и повторное использование статических моделей: практически применимая архитектура должна содержать доверенную коллекцию ранее обработанных приложений, репозиторий их статических моделей и индекс предварительного отбора, чтобы не извлекать все признаки заново для уже известных объектов при каждом новом сравнении.

7. Правильное место слоя формирования заключения: прикладной слой может использовать оценку сходства и материалы интерпретации для подготовки заключения, но не должен смешиваться с ядром анализа сходства.

Далее, после констатации этих принципов, покажем, как они раскладываются на составные подсистемы. Такое структурное представление приведено на рис. 2.

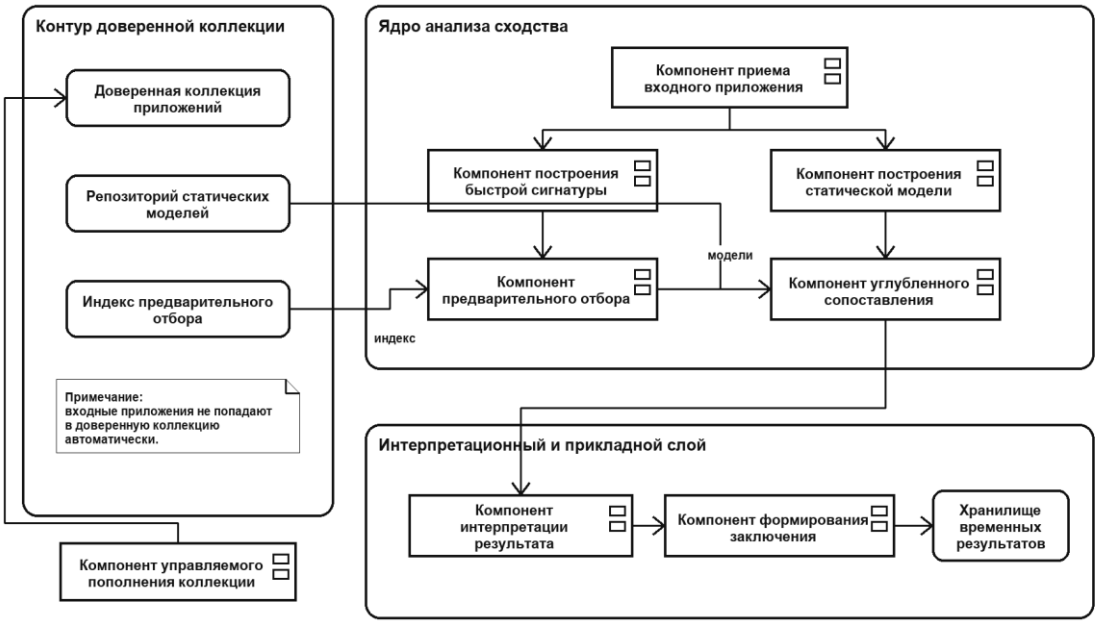


Рис. 2. UML-диаграмма компонентов системы многоуровневого анализа сходства приложений для платформы Android.

На рис. 2 ядро анализа разделено на подсистемы приема входного приложения, построения статической модели, формирования быстрой сигнатуры, предварительного отбора кандидатов, углубленного сопоставления, интерпретации результата и формирования заключения. Отдельно выделены доверенная коллекция приложений, репозиторий статических моделей, индекс предварительного отбора, временное хранилище результатов и управляемое пополнение доверенной коллекции. В настоящей работе доверенная коллекция понимается как курируемый оператором набор APK-артефактов и заранее построенных моделей, пригодных для повторного использования в поисковом контуре. Такое представление позволяет развести процессный уровень и уровень составных частей системы: рис. 1 отвечает на вопрос о ходе анализа, а рис. 2 – на вопрос о составе подсистем и их зависимостях.

Основные этапы многоуровневого анализа можно свести к следующим положениям.

1. Предварительный отбор: по быстрой сигнатуре входного приложения и индексу доверенной коллекции формируется упорядоченный список кандидатов.

2. Углубленное сопоставление: для входного приложения и отобранных кандидатов вычисляется нормированная оценка сходства на уровне статических моделей.

3. Интерпретация результата: по данным углубленного сопоставления фиксируются источники сходства и различия, включая вклад библиотечного слоя.

4. Формирование заключения: на основе оценки сходства и структурированной интерпретации готовится заключение для эксперта.

Формализованное описание двухэтапного контура, согласованное с обозначенными уровнями, дано в Алгоритме 1. Вход: входное приложение a, индекс коллекции I, репозиторий статических моделей R, оператор моделирования M, ограничение k. Выход: множество кандидатов K, оценки сходства S, структурированные интерпретации E, заключения L, служебные технические статусы F и нормализованные причины отказа G.

Алгоритм 1. Двухэтапный контур многоуровневого анализа сходства приложений для платформы Android

```
ANALYZE_SIMILARITY(a, I, R, M, k)
1   X[a] ← BuildStaticModel(a, M)
2   if X[a] = NIL
3       then return FAIL, input_model_failed
4   Z[a] ← BuildFastSignature(a)
5   if Z[a] = NIL
6       then return FAIL, fast_signature_failed
7   ▷ Предварительный отбор по доверенной коллекции
8   q ← SearchIndex(Z[a], I)
9   K ← SelectTopCandidates(q, k)
10  for each b ∈ K
11      do Y[b] ← LoadCollectionModel(R, b)
12         if Y[b] = NIL
13             then F[b] ← FAIL
14                  G[b] ← candidate_model_missing
15             else S[a, b] ← ComputeSimilarity(X[a], Y[b])
16                  E[a, b] ← BuildInterpretation(X[a],Y[b],S[a,b])
17                  L[a, b] ← BuildConclusion(S[a, b], E[a, b])
18  return K, S, E, L, F, G
```

ЭКСПЕРИМЕНТАЛЬНЫЙ КОНТУР

Экспериментальный контур нужен не для демонстрации одной итоговой цифры, а для раздельной проверки различных слоев предлагаемой архитектуры. Поэтому отдельно рассматриваются предварительный отбор кандидатов, углубленное сопоставление, интерпретация результата и устойчивость вычислительного контура. Такое разбиение позволяет проследить, на каком именно этапе возникает ограничение или, наоборот, наблюдается улучшение результата. Все результаты этого раздела следует рассматривать как пилотные наблюдения на локальном наборе данных, а не как завершенную валидацию метода на больших коллекциях.

В текущем пилотном цикле экспериментальный контур был разделен на основной и граничный. В качестве основного пилотного набора использовался локальный набор пар dataset-v2-core. Под этим обозначением понимается набор из пяти пар APK-файлов с зафиксированным происхождением, доступными локальными артефактами, анализируемым байткодом и успешным прохождением исходного байткод-ориентированного режима без специальных обходов. В его состав входят три пары с общим происхождением (P01–P03) и две контрольные пары без общего происхождения (P05, P06).

Отдельно от основного набора рассматривался граничный контур устойчивости. В него включались случаи, которые не следует использовать ни для калибровки рабочего порога, ни для итоговой оценки качества на основном наборе. Пары P11 и P12 были нужны для разведения двух разных ограничений: риска чрезмерного подавления полезного сигнала на шаге библиотечной редукции и риска нестабильного построения модели на паре с общим происхождением.

До прямого сопоставления базового и улучшенного режимов была восстановлена воспроизводимость исходного байткод-ориентированного контура: после исправлений совместимости удалось повторить пять исторических значений сходства на локальном пилотном материале. Для оценки разделимости классов на основном наборе использовалась разность между минимальной оценкой сходства в группе пар с общим происхождением и максимальной оценкой в контрольной группе без общего происхождения. Чем больше эта разность, тем лучше режим разделяет похожие и непохожие пары.

На основном наборе оба режима завершили анализ всех пяти пар без служебного технического статуса. Ключевые результаты этого сопоставления приведены ниже.

1. Основной пилотный набор: восстановлены пять исторических значений, а dataset-v2-core включает пять пар APK-файлов: три пары с общим происхождением и две контрольные пары без общего происхождения. Этого достаточно для сопоставления базового и улучшенного режимов на пилотном материале, но недостаточно для статистически полной валидации.

2. Разделение классов: в базовом режиме минимальная оценка в группе пар с общим происхождением оказалась ниже максимальной оценки в контрольной группе на 0.041804, тогда как в улучшенном режиме она превысила максимальную оценку контрольной группы на 0.135935. На данном пилотном наборе это наблюдение указывает на более отчетливое разделение похожих и непохожих пар, но еще не доказывает превосходства режима на широком корпусе.

3. Порог 0.15: при этом пороге базовый режим дал три верно распознанные близкие пары, один ложный сигнал сходства и один верно распознанный контрольный случай, тогда как улучшенный режим сохранил все

три верно распознанные близкие пары и устранил ложный сигнал сходства. Покрытие слоя интерпретации одновременно выросло с 4/5 до 5/5. В текущей работе порог 0.15 используется как иллюстративная рабочая граница пилотного прогона, а не как окончательно валидированная граница решения.

4. Граничные случаи P11 и P12: для P11 после библиотечной редукции значение сходства снизилось с 0.2 до 0, что указывает на риск чрезмерного подавления полезного сигнала. Для P12 в стандартной конфигурации фиксировался статус FAIL, тогда как в последовательной конфигурации оценка изменилась с 0.80543 до 0.276018. Это показывает, что в граничных случаях текущий прототип пока не полностью разделяет ограничения метода и ограничения реализации.

ОБСУЖДЕНИЕ

Предлагаемая постановка полезна не только для задач анализа приложений для платформы Android как таковых, но и для более общего контекста работы с большими цифровыми коллекциями приложений. В тематике электронных библиотек существенна не столько рекомендация в узком пользовательском смысле, сколько задача поиска, сопоставления и интерпретации сходства между элементами цифровой коллекции. В этом отношении предлагаемый контур анализа сходства может рассматриваться как инструмент интеллектуального поиска по коллекции приложений, где ранжирование кандидатов образует первый этап работы, а исследовательская ценность возникает на уровне интерпретации результата и воспроизводимости оснований для вывода.

С методической точки зрения необходимо удерживать границы текущих утверждений. Настоящая работа не доказывает масштабируемость анализа на больших рыночных коллекциях, не вводит окончательно валидированный многоуровневый метод и не утверждает завершенность слоя формирования заключения. Ее вклад состоит в последовательном расширении предшествующей байткод-ориентированной линии наших работ: формализуется функция сходства приложений по статическим признакам, вводится статическая модель приложения как рабочий объект анализа, а экспериментальная проверка выносится в самостоятельный контур, в котором

технические ограничения текущего прототипа фиксируются отдельно от содержательного результата сравнения.

Научная новизна настоящей работы определяется четырьмя положениями. Во-первых, функция сходства приложений формулируется для многоуровневого набора статических признаков, а не только для байткодного слоя. Во-вторых, статическая модель приложения вводится как явный рабочий объект анализа, включающий манифест, ресурсы, APK-внутренние метаданные и библиотечные зависимости. В-третьих, архитектура анализа строится как двухконтурная схема работы с доверенной коллекцией, где предварительный отбор и углубленное сопоставление разведены по ролям. В-четвертых, экспериментальный контур отделяет пилотный основной набор от граничных случаев и не смешивает ограничения текущего прототипа с содержательным результатом сравнения.

ЗАКЛЮЧЕНИЕ

Сформулирована задача многоуровневого анализа сходства приложений для платформы Android по статическим признакам, введена статическая модель приложения как рабочий объект сравнения, предложена архитектура соответствующего анализа и уточнен экспериментальный контур его проверки. Показано, что переход от базового режима, ориентированного только на итоговый показатель и только на байткод, к многоуровневому контуру обусловлен природой самого объекта исследования: приложение для платформы Android содержит значимые сигналы сходства не только в коде, но и в манифесте, ресурсах, метаданных и библиотечных зависимостях.

Предложенная архитектура исходит из разделения предварительного и углубленного анализа, обязательной интерпретации результата, явного учета влияния общих библиотечных компонентов и отдельной фиксации технических ограничений вычислительного контура. Отдельно подчеркнем, что слой формирования заключения может быть естественной прикладной надстройкой над слоем интерпретации, но не должен подменять собой ядро анализа сходства.

Результаты пилотного экспериментального цикла согласуются с гипотезой о полезности явного учета влияния общих библиотечных компонентов и

раздельной фиксации технических ограничений текущего прототипа, однако не дают оснований для обобщающих утверждений о завершенной валидации архитектуры на больших коллекциях. Дальнейшая работа связана с реализацией и сравнительной оценкой отдельных слоев многоуровневого контура, расширением многоуровневой статической модели приложения, повышением устойчивости вычислительного контура и построением воспроизводимого экспериментального контура на основных и граничных наборах данных.

СПИСОК ЛИТЕРАТУРЫ

1. *Li L. et al.* Understanding Android App Piggybacking: A Systematic Study of Malicious Code Grafting // IEEE Transactions on Information Forensics and Security. 2017. Vol. 12, No. 6. P. 1269–1284. https://doi.org/10.1109/TIFS.2017.2656460

2. *Петров В.В.* Система автоматизации численной оценки сходства Android-приложений // Научный сервис в сети Интернет: труды XXV Всероссийской научной конференции (18–21 сентября 2023 г., онлайн). М.: ИПМ им. М.В. Келдыша, 2023. С. 283–297. https://doi.org/10.20948/abrau-2023-33

3. *Петров В.В.* Система автоматизации численной оценки сходства Android-приложений // Электронные библиотеки. 2024. Т. 27, № 3. С. 336–365. https://doi.org/10.26907/1562-5419-2024-27-3-336-365

4. *Petrov V.V.* Automated System for Numerical Similarity Evaluation of Android Applications // Automatic Documentation and Mathematical Linguistics. 2024. Vol. 58 (Suppl. 3). P. 131–142. https://doi.org/10.3103/S0005105525700207

5. *Cesare S., Xiang Y.* Software Similarity and Classification. London: Springer, 2012. 88 p. https://doi.org/10.1007/978-1-4471-2909-7

6. *Desnos A.* Android: Static Analysis Using Similarity Distance // Proc. of the 45th Hawaii International Conference on System Sciences. 2012. P. 5394–5403. https://doi.org/10.1109/HICSS.2012.114

7. *Rastogi V., Chen Y., Jiang X.* DroidChameleon: Evaluating Android Anti-Malware Against Transformation Attacks // Proc. of the 8th ACM SIGSAC. 2013. P. 329–334. https://doi.org/10.1145/2484313.2484355

8. *Zhauniarovich Y. et al.* FSquaDRA: Fast Detection of Repackaged Applications // Data and Applications Security and Privacy XXVIII. 2014. P. 130–145. https://doi.org/10.1007/978-3-662-43936-4_9

9. *Li L., Bissyande T. F., Klein J.* Rebooting Research on Detecting Repackaged Android Apps // IEEE Transactions on Software Engineering. 2021. Vol. 47, No. 4. P. 676–693. https://doi.org/10.1109/TSE.2019.2901679

10. *Li L., Bissyande T. F., Klein J.* SimiDroid: Identifying and Explaining Similarities in Android Apps // 2017 IEEE Trustcom/BigDataSE/ICESS. 2017. P. 136–143. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.230

11. *Backes M., Bugiel S., Derr E.* Reliable Third-Party Library Detection in Android and Its Security Applications // Proc. of the ACM Conf. on Computer and Comm. Security. 2016. P. 356–367. https://doi.org/10.1145/2976749.2978333

12. *Li M. et al.* LibD: Scalable and Precise Third-Party Library Detection in Android Markets // Proc. of the 39th Int. Conf. on Software Engineering. 2017. P. 335–346. https://doi.org/10.1109/ICSE.2017.38

13. *Huang J. et al.* Scalably Detecting Third-Party Android Libraries With Two-Stage Bloom Filtering // IEEE Transactions on Software Engineering. 2023. Vol. 49, No. 4. P. 2272–2284. https://doi.org/10.1109/TSE.2022.3215628

MODEL AND ARCHITECTURE OF MULTI-LEVEL SIMILARITY ANALYSIS OF ANDROID APPLICATIONS BASED ON STATIC FEATURES

V. V. Petrov [0009-0004-4213-7328]

Kazan Federal University, Kazan, Russia

valeryvpetrov.itis@gmail.com

Abstract

The paper addresses the problem of multi-level similarity analysis of Android applications based on static features in digital application collections. Such collections may contain duplicates, forks, repackaged builds, and other modified variants; malicious payloads are treated as a special case of modification rather than as a synonym of repackaging. The paper formulates a similarity function for Android applications, introduces a static application model as the working object of comparison, and presents a multi-level pipeline that separates candidate screening, in-depth pairwise analysis, result interpretation, and a decision layer. Meaningful

similarity signals are sought not only in classes.dex bytecode, but also in AndroidManifest.xml, resources, APK-internal metadata, and library dependencies. A numerical similarity score is computed only when static models are built successfully; otherwise the pipeline records a dedicated technical failure status together with a normalized failure reason. Preliminary evidence is reported on a local pilot set of five core pairs and two boundary cases. These results indicate that explicit handling of shared library code may improve interpretability, but they do not yet constitute a full validation of the proposed architecture on large collections.

Keywords: *Android applications, static analysis, program similarity analysis, search for modified variants, repackaged applications, library dependencies, result interpretation, digital collections of applications.*

REFERENCES

1. *Li L. et al.* Understanding Android App Piggybacking: A Systematic Study of Malicious Code Grafting // IEEE Transactions on Information Forensics and Security. 2017. Vol. 12, No. 6. P. 1269–1284. https://doi.org/10.1109/TIFS.2017.2656460

2. *Petrov V.V.* System of Automated Numerical Similarity Evaluation of Android Applications // Nauchnyi servis v seti Internet: Trudy XXV Vserossiiskoi nauchnoi konferentsii. 2023. P. 283–297. https://doi.org/10.20948/abrau-2023-33

3. *Petrov V.V.* System of Automated Numerical Similarity Evaluation of Android Applications // Russian Digital Libraries Journal. 2024. Vol. 27, No. 3. P. 336–365. https://doi.org/10.26907/1562-5419-2024-27-3-336-365

4. *Petrov V.V.* Automated System for Numerical Similarity Evaluation of Android Applications // Automatic Documentation and Mathematical Linguistics. 2024. Vol. 58 (Suppl. 3). P. 131–142. https://doi.org/10.3103/S0005105525700207

5. *Cesare S., Xiang Y.* Software Similarity and Classification. London, Springer, 2012. 88 p. https://doi.org/10.1007/978-1-4471-2909-7

6. *Desnos A.* Android: Static Analysis Using Similarity Distance // Proc. of the 45th Hawaii International Conference on System Sciences. 2012. P. 5394–5403. https://doi.org/10.1109/HICSS.2012.114

7. *Rastogi V., Chen Y., Jiang X.* DroidChameleon: Evaluating Android Anti-Malware Against Transformation Attacks // Proc. of the 8th ACM SIGSAC. 2013. P. 329–334. https://doi.org/10.1145/2484313.2484355

8. *Zhauniarovich Y. et al.* FSquaDRA: Fast Detection of Repackaged Applications // Data and Applications Security and Privacy XXVIII. 2014. P. 130–145. https://doi.org/10.1007/978-3-662-43936-4_9

9. *Li L., Bissyande T. F., Klein J.* Rebooting Research on Detecting Repackaged Android Apps // IEEE Transactions on Software Engineering. 2021. Vol. 47, No. 4. P. 676–693. https://doi.org/10.1109/TSE.2019.2901679

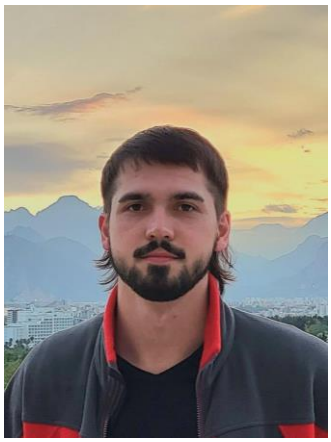
10. *Li L., Bissyande T. F., Klein J.* SimiDroid: Identifying and Explaining Similarities in Android Apps // 2017 IEEE Trustcom/BigDataSE/ICESS. 2017. P. 136–143. https://doi.org/10.1109/Trustcom/BigDataSE/ICESS.2017.230

11. *Backes M., Bugiel S., Derr E.* Reliable Third-Party Library Detection in Android and Its Security Applications // Proc. of the ACM Conf. on Computer and Comm. Security. 2016. P. 356–367. https://doi.org/10.1145/2976749.2978333

12. *Li M., Wang W., Wang P. et al.* LibD: Scalable and Precise Third-Party Library Detection in Android Markets // Proc. of the 39th International Conference on Software Engineering. 2017. P. 335–346. https://doi.org/10.1109/ICSE.2017.38

13. *Huang J., Zhang Y., Tan H. et al.* Scalably Detecting Third-Party Android Libraries With Two-Stage Bloom Filtering // IEEE Transactions on Software Engineering. 2023. Vol. 49, No. 4. P. 2272–2284. https://doi.org/10.1109/TSE.2022.3215628

СВЕДЕНИЯ ОБ АВТОРЕ



ПЕТРОВ Валерий Владимирович – магистр программной инженерии, аспирант Института информационных технологий и интеллектуальных систем Казанского федерального университета. Научные интересы: анализ сходства приложений для платформы Android, статический анализ программ, интерпретируемые методы сравнения программных артефактов, воспроизводимые исследовательские контуры.

Valery Vladimirovich PETROV – Master of Software Engineering, postgraduate student at the Institute of Information Technology and Intelligent Systems, Kazan Federal University. Research interests: Android application similarity, static program analysis, interpretable comparison of software artifacts, reproducible research workflows**.**

email: valeryvpetrov.itis@gmail.com

ORCID: 0009-0004-4213-7328

Материал поступил в редакцию 20 марта 2026 года

УДК 81'32+81'33

К ВОПРОСУ О ПРЕДСТАВЛЕНИИ СИНТАГМАТИЧЕСКИХ ОТНОШЕНИЙ МОРФЕМ В ВЕКТОРНЫХ ЯЗЫКОВЫХ МОДЕЛЯХ

Д. К. Родионова[1] [0009-0004-6296-8532], О. А. Митрофанова[2] [0000-0002-3008-5514]

[1, 2]*Санкт-Петербургский государственный университет,*
г. Санкт-Петербург, Россия

[1]*НИИ Исследовательская лаборатория им. П. Л. Чебышева,*
г. Санкт-Петербург, Россия

[1]rodionowadarja@yandex.ru, [2]o.mitrofanova@spbu.ru

Аннотация

В работе рассмотрено представление семантической структуры производных слов в языковых моделях, учитывающее внутрисловные синтагматические отношения между словообразовательными морфемами. Эксперименты проводились с привлечением морфемных моделей НейроКРЯ, а также моделей fastText и ruRoBERTa. Проверена гипотеза о композициональности производных слов, представляемых в виде агрегированных векторов морфем, а также выполнено сравнение представлений семантических отношений с помощью морфемных векторов fastText и стандартных векторов подслов в модели ruRoBERTa. Полученные результаты указывают на умеренную чувствительность векторов fastText к синтагматическим связям между морфемами и словообразовательным типам. Установлено также что агрегация морфемных векторов в fastText улучшает регистрацию семантических отношений между словами, связанными словообразовательными отношениями, по сравнению с агрегацией векторов подслов в модели ruRoBERTa.

Стандартные токенизаторы BPE (Byte-Pair Encoding) и WordPiece, применяемые в моделях семейства Transformer, являются слабоинтерпретируемыми в отношении языковых данных, поскольку в них сегменты слов не всегда соответствуют морфемам. Исследовательская проблема состоит в необходимости оценки того, в какой мере современные языковые модели способны регистрировать лингвистические признаки, характеризующие отношения производных слов в словообразовательных гнездах.

В работе оценена способность предсказывающих моделей распределенных векторных вложений воспроизводить синтагматические связи между морфемами внутри производных слов и на уровне словообразовательных гнезд в русском языке.

Полученные результаты стимулируют разработку нейросетевых архитектур, учитывающих синтагматические отношения между морфемами, совершенствование морфемных токенизаторов и их интеграцию в языковые модели.

***Ключевые слова**: языковая модель, морфемный анализ, словообразовательные способы, композициональность.*

ВВЕДЕНИЕ

На сегодняшний день ни одна задача обработки естественного языка не обходится без применения методов векторизации текстовых данных и интеграции больших языковых моделей в лингвистические процессоры. Современные подходы к анализу текстов пользуются большой популярностью благодаря появлению вычислительных ресурсов, позволяющих обработать большие объемы данных, что обеспечивает высокое качество моделей и верификацию результатов. В то же время все больше вопросов возникает в связи с интерпретируемостью внутренних представлений моделей и их соответствием языковым единицам различных уровней, в том числе морфем [1]. Токенизаторы классов BPE и WordPiece, используемые в моделях семейства Transformer, являются слабоинтерпретируемыми, поскольку выделяют сегменты слов, не всегда соответствующие морфемам. Во многих работах было показано положительное влияние морфемной токенизации на качество генерации текстов с использованием приемов перефразирования, суммаризации, упрощения в языках с богатыми словообразованием и словоизменением (русский, белорусский, сербский, чешский, финский, эстонский и т. д.). Кроме того, морфемный анализ может положительно влиять на качество морфологической аннотации текстов и генерации морфологических форм слов [2–6]. Несмотря на это, исследование внутренней структуры слова в русскоязычных языковых моделях недостаточно широко представлено в публикациях.

Цель настоящего исследования состояла в оценке способности предсказывающих моделей распределенных векторных вложений воспроизводить синтагматические связи между морфемами внутри производных слов и на уровне словообразовательных гнезд в русском языке. В ходе исследования проверялась гипотеза о композициональности производных слов при агрегации морфемных векторов.

В статье дан обзор аналогичных исследований, описан исследовательский набор данных, обоснован выбор моделей fastText и ruRoBERTa, представлены способы агрегации векторов производных слов, а также проведены анализ результатов сравнения агрегированных векторов для исследовательского набора данных в моделях и оценка способности моделей fastText и ruRoBERTa воспроизводить семантические отношения внутри словообразовательных гнезд. Полученные результаты подтверждают перспективность учета границ морфем в разработке токенизаторов для языковых моделей.

БЛИЗКИЕ ПО ТЕМАТИКЕ ИССЛЕДОВАНИЯ

Вычислительные аспекты нашего исследования согласуются с тенденциями развития языкового моделирования как задачи искусственного интеллекта, в то время как лингвистические основания связаны с особым направлением в формальной лингвистике, а именно с генеративной морфологией, применяющей аппарат формальных грамматик в описании процессов деривации [7], и теорией гипосинтаксиса, объясняющей природу синтагматических отношений между морфемами [8]. Для русского языка, для которого характерны богатая морфологическая система, развитые словоизменение и словообразование, особо важно, что производные слова обладают дискретной структурой как в плане выражения, так и в плане содержания. Значение производного слова возникает в результате воздействия словообразовательного аффикса на производящую основу. Это дает основания считать внутрисловные связи между морфемами разновидностью синтаксических отношений, поэтому, производное слово может рассматриваться как аналог словосочетания и предложения [9–12]. Интеграция генеративного и традиционного подходов к описанию семантики производного слова реализована в деривационных моделях, использующих падежно-ролевой подход [13–15]. Тем самым в указанных работах рассмотрена

проблема композициональности семантики производных слов, но вне задачи обучения и применения языковых моделей.

С возможностью введения в корпусы текстов словообразовательной разметки (в частности, в НКРЯ) и учета деривационных связей в компьютерных тезаурусах типа WordNet задача моделирования словообразовательных отношений стала более реалистичной. В условиях ограниченных обучающих данных применимы обучение без учителя и статистические подходы, в частности, в инструменте Morfessor [16] реализован алгоритм вероятностной сегментации слов на морфемы, адаптируемый к различным языкам (финский, турецкий, эстонский, русский и т. д.).

При наличии обучающих данных высокие результаты обеспечиваются алгоритмами глубинного обучения. В частности, для русского языка существует группа нейросетевых моделей, обученных под задачу морфемной сегментации и классификации: CNN, LSTM, GBDT, BERT [2–6]. Нейросетевая классификация морфем состоит в присвоении части слова одной из специальных меток: префикса, корня, суффикса, окончания и т. д.

В работах [2, 3] представлены программный комплекс RussianMorphParsing [17] и набор данных RuMorphs-Lemmas, в [6] – инструмент и модели ruMorpheme [18], в серии публикаций [4, 5] и репозитории Neuromodels [19] – нейросетевые модели семейства BERT и словари, используемые в проекте НейроКРЯ. Недавно были представлены исследования, в которых рассматривались токенизаторы для моделей семейства Transformer, основанные на сегментации слов на морфемы [20–22]. Было показано, что благодаря такой стратегии они помогают повысить качество в решении различных лингвистических задач в отличие от обычных BPE-токенизаторов, которые при сегментации слов не учитывают границы морфем.

Несмотря на разнообразие решений задачи морфемной сегментации и классификации, до сих пор не решен вопрос о представлении синтагматических связей между морфемами в производных словах и отношений производности в словообразовательных гнездах. В настоящей работе предложено решение этих проблем.

ЭКСПЕРИМЕНТ

Данные

В качестве источника данных для серии экспериментов были использованы «Школьный словарь строения слов русского языка» З. А. Потихи объемом около 25 тыс. слов [23] и «Морфемно-орфографический словарь русского языка» А. Н. Тихонова объемом около 100 тыс. слов [24]. При отборе материала из этих источников учитывалась частотность целевых слов, а также репрезентативность их словообразовательных гнезд с точки зрения разнообразия словообразовательных способов. Мы также учитывали возможные разночтения в вариантах морфемной сегментации, представленных в разных источниках. По этим критериям были выбраны семь словообразовательных гнезд для существительных: *свет*, *лес*, *вода*, *дом*, *слово*, *земля* и *снег*. Объем гнезд для каждого производящего слова составлял примерно 50 лексических единиц. Общее число производных составляет более 350 лексических единиц. В каждом из гнезд представлены префиксально-суффиксальный, суффиксальный, префиксальный, сложно-суффиксальный словообразовательные способы, а также сложение основ (табл. 1), что позволило исследовать чувствительность языковых моделей к словообразовательным способам.

Табл. 1. Данные по словообразовательным гнездам.

Словообразовательный способ	СВЕТ	ЛЕС	ВОДА	ДОМ	СЛОВО	ЗЕМЛЯ	СНЕГ
Префиксно-суффиксальный	13	10	9	11	9	13	10
Суффиксальный	10	11	9	12	12	13	12
Префиксальный	6	5	1	0	1	2	0
Сложение основ	10	11	13	11	10	11	10
Сложно-суффиксальный	10	10	19	11	12	11	11
Общее количество	49	47	51	45	44	50	43

Умеренные объемы данных обусловлены тем, что на данном этапе исследования отсутствует такой инструмент, с помощью которого можно автоматизировать процесс сбора производных слов из предложенных выше словарей для составления гнезд. На величину гнезда влияет также исключение слов, которые

при одинаковом словообразовательном способе имеют различные окончания (например, в паре *светлый – светлая* оставляем первое слово).

Модели

Нейросетевые морфемные модели CNN, LSTM, GBDT, BERT в комбинации со словарными данными позволяют достичь при решении задачи морфемной сегментации значений F-меры на уровне 0.99. Модели MorphBERTa, разработанные НейроКРЯ [19], показывают на сегодняшний день наилучшие результаты в задачах определения морфемных границ и назначения морфемных меток. Однако, несмотря на свои преимущества, они имеют некоторые ограничения.

Во-первых, модели MorphBERTa не обучались для задачи распознавания границ предложений и не адаптированы для разрешения неоднозначности некоторых грамматических характеристик слов в контексте (например, словоформа *пора* в зависимости от синтаксической структуры предложения может быть аннотирована либо как предикативное наречие, либо как существительное).

Во-вторых, модели MorphBERTa не предназначены для распознавания словообразовательных способов (например, *учащийся* прич. → сущ.). Эти наблюдения требуют пересмотра исследовательского набора данных при подготовке экспериментов.

Для проверки гипотезы о композициональности производных слов при агрегации морфемных векторов мы рассмотрели группу моделей из семейства fastText [25], которые не были дообучены для обработки морфемной информации. Благодаря обучению на *n*-граммах (последовательностях графем внутри слов) модели fastText способны распознавать слова, отсутствующие в обучающих данных, и делать предсказания в отношении несловарных слов. Из предобученных моделей для русского языка были использованы *geowac_lemmas* и *geowac_tokens* с размером окна 5 и размерностью вектора 300 [26]. Дополнительно был проведен эксперимент с моделями Transformer для оценки способности моделей воспроизводить семантические отношения внутри словообразовательных гнезд. Из семейства BERT мы выбрали ruRoBERTa-large [27, 28] как альтернативу составной модели fastText и MorphBERTa, не содержащую информа-

цию о морфемном членении и разметке. В качестве токенов ruRoBERTa-large кодирует подслова. Токенизация проводится с помощью алгоритма BPE, который разбивает входные слова на подстроки и ранжирует их таким образом, что в словаре модели сохраняются наиболее частотные последовательности символов, которые далеко не всегда соответствуют морфемам.

Методы и метрики

В экспериментах были использованы следующие методы агрегации векторов производных слов. Для каждого слова в словообразовательном гнезде были сформированы три вектора: вектор производного слова, вектор из композиции морфем, а также вектор основы. Для вектора композиции агрегация проводилась одним из трех способов: это усреднение, сумма и выбор максимальной координаты. Далее вычисляли следующие косинусные метрики, которые затем собирались для каждого гнезда в отдельные выборки:

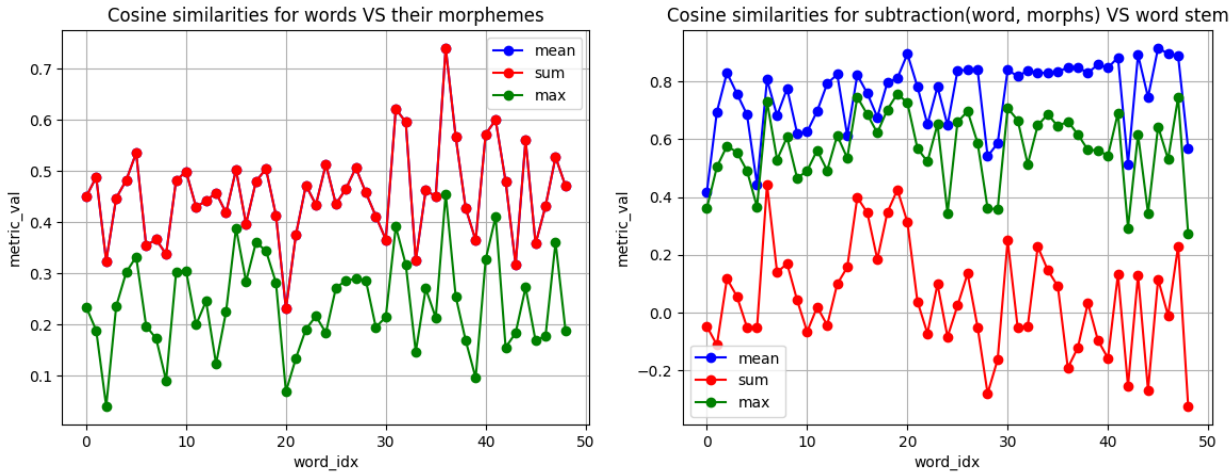
КМ-1: $\mathrm{cos}ine\big(w, aggr(m_i)\big)$;
КМ-2: $cosine(w\ -\ aggr(m_i), s)$, где
w – вектор слова, s – вектор основы слова,
$\{m_i\}$ – морфемный ряд, *aggr* = [*'mean'*, *'sum'*, *'max'*].

На первом этапе сравнивали способы агрегации векторов морфем в паре моделей fastText, из которых модель *geowac_tokens* была обучена на словоформах, а *geowac_lemmas* – на леммах. По данным, полученным по каждому из словообразовательных гнезд, выполняли дисперсионный анализ и его аналоги (тесты Краскела и Уелча) с целью проверки соотношения между словообразовательными способами и значениями косинусной метрики, а также выбора тех словообразовательных способов, которые лучше других представлены в моделях. Данный анализ проводился со значением *p*-value, равным 5%. Аналогичные шаги были также выполнены и для модели ruRoBERTa-large.

РЕЗУЛЬТАТЫ

Проверка гипотезы о композициональности производных слов при агрегации морфемных векторов

В ходе первого эксперимента было установлено, что обе модели fastText, обученные на словоформах и леммах, при использовании агрегации морфемных векторов методом усреднения дают наилучшие результаты. При этом значение косинусной метрики в целом не превышает 0.5, что означает умеренную степень близости между вектором слова и агрегированным вектором морфем. На рис. 1 представлены графики изменений значения косинусной метрики в словообразовательном гнезде слова *свет*.

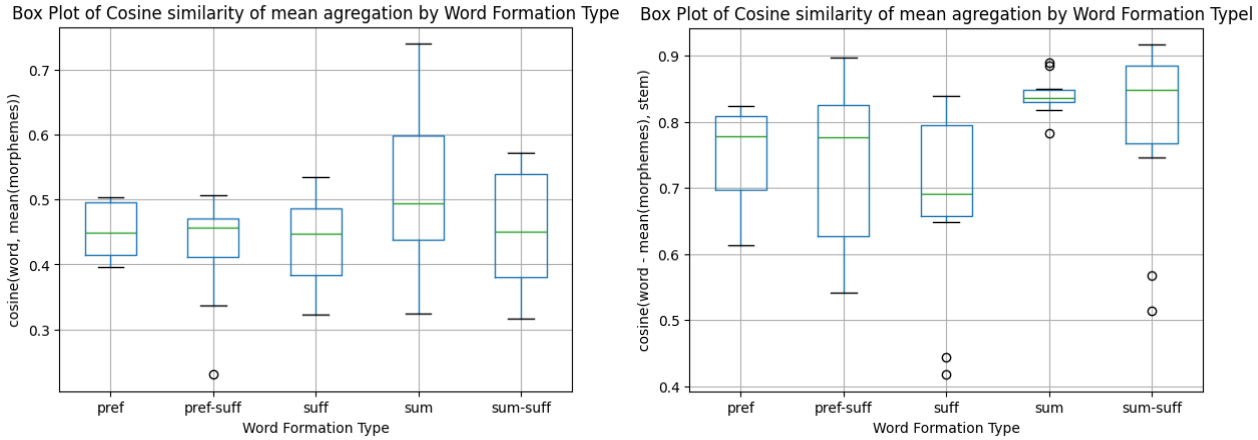


(а) (б)

Рис. 1. Косинусные значения для агрегаций векторов морфем в словообразовательном гнезде *свет*: а) КМ-1, б) КМ-2.

Было исследовано соотношение между словообразовательными способами и значениями косинусной метрики для агрегированных векторов. Отдельные словообразовательные способы и их значения метрики КМ-1 для словообразовательных гнезд представлены на ящиках с усами (рис. 2а – сравниваются векторы морфем, рис. 2б – сравниваются вектор основы и разность вектора слова и сводного вектора морфем). Следует заметить, что словообразовательный способ, связанный со сложением основ слов, показывает самые высокие результаты по метрике КМ-2 в случае агрегации морфемных векторов методом

усреднения. Это означает, что модели fastText могут обрабатывать многоосновные слова, имеющие слитное написание (например, *золотоискатель*, *Роспотребнадзор*). Однако такая закономерность не наблюдается для метрики КМ-1. Например, для слова *вода* наиболее высокие косинусные метрики соответствуют группе суффиксальной словообразовательной модели.



(а) (б)

Рис. 2. Распределение словообразовательных способов для словообразовательного гнезда *свет:* а) КМ-1, б) КМ-2.

Для проверки влияния способа словообразования на значение косинусной метрики может быть применен дисперсионный анализ при условии, что распределение исследуемых данных подчиняется нормальному закону. Если в выборках обнаруживались выбросы, часть из них исключалась, если они возникли вследствие ошибок модели НейроКРЯ. Например, для производящего слова *словарь* модель вернула морфемный ряд, состоящий только из одного корня *словарь*, что не соответствует правильному разбору, в котором выделяется суффикс *-арь*.

Для всех словообразовательных гнезд у fastText значения косинусных метрик КМ-2 оказались выше, чем КМ-1 (табл. 2). Более того, по результатам статистического сравнения двух моделей fastText для КМ-2 лучше всего подходит модель, обученная на леммах. Тем самым гипотеза о композициональности подтверждается при сравнении основы слова с разностью вектора слова и сводного вектора морфем. Однако, если исходить из значений косинусной метрики, связь

между вектором слова и агрегацией векторов морфем менее очевидна. Для определения сходства между агрегированными морфемами и словоизменительными аффиксами (прежде всего, флексии) рассчитывалась косинусная близость их векторов. Оказалось, что при усреднении векторов морфем выделялось только 30% флексий, для которых значения косинусной метрики были выше 0.7. Это указывает на слабую взаимосвязь между агрегированными морфемами и флексиями, а также позволяет предположить, что агрегированный вектор способен нести в себе более сложную информацию, чем вектор отдельной морфемы. Однако метрики модели ruRoBERTa-large имеют иные показатели: здесь значения КМ-1 выше, чем значения КМ-2, более того, КМ-1 у ruRoBERTa-large значительно выше КМ-1 у fastText. В свою очередь, это может говорить о том, что модель семейства BERT лучше распознает словообразовательные признаки, чем fastText. С другой стороны, связь между словом и композицией его морфем в КМ-1, оцениваемая через косинусную метрику, не является сильной. Иными словами, мы не можем в этом случае ни подтвердить, ни опровергнуть гипотезу композициональности.

Табл. 2. Средние значения косинусной метрики для двух экспериментов.

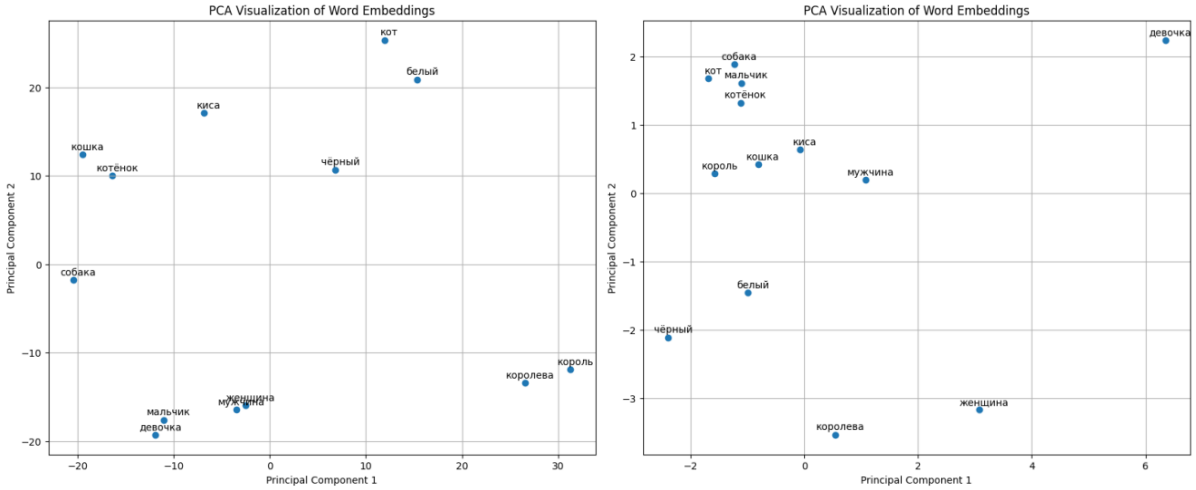
Модели		ВОДА	ЗЕМЛЯ	СВЕТ	ЛЕС	ДОМ	СЛОВО	СНЕГ
Объем гнезда		51	50	49	47	45	44	43
fastText	КМ-1, mean	0.423	0.526	0.456	0.485	0.46	0.479	0.512
geowac_lemma	КМ-2, mean	**0.797**	**0.71**	**0.757**	**0.723**	**0.69**	**0.668**	**0.743**
ruRoBERTa-large	КМ-1, mean	**0.646**	**0.708**	**0.696**	**0.707**	**0.703**	**0.682**	**0.741**
subword aggregation = mean	КМ-2, mean	0.387	0.371	0.384	0.379	0.332	0.331	0.364

Итак, результаты проведенного эксперимента подтверждают, что предсказывающие модели распределенных векторных вложений недостаточно полно воспроизводят синтаксические отношения между морфемами и поэтому не могут представлять композиционную семантику производных слов при агрегации

морфемных векторов. Таким образом, наша гипотеза не подтверждена. Полученные результаты стимулируют исследования, направленные на поиск нейросетевых архитектур, которые позволили бы обучить искомые модели. На возможность решения такой задачи указывает и то, что в языковых моделях воспроизводятся синтагматические отношения внутри предложений. Это означает, что при наличии соответствующей разметки на уровне морфемики и морфологии модели смогут интерпретировать подобные связи и внутри слова.

Оценка способности моделей воспроизводить семантические отношения внутри словообразовательных гнезд

Дополнительно был проведен второй эксперимент, направленный на сравнение моделей fastText и ruRoBERTa в задаче установления семантических связей между словами в словообразовательных гнездах с опорой на векторы подслов и морфем. Результаты представлены на рис. 3. Очевидно, что для модели fastText родовидовые отношения и дифференциация по признаку пола являются однонаправленными и более близкими, чем в модели ruRoBERTa (ср. векторы для лемм *кошка*, *котенок; кот* и *киса* более компактно расположены в fastText и более рассредоточены в пространстве ruRoBERTa). Значит, модель ruRoBERTa регистрирует семантическую близость векторов слов без учета их морфемного состава и словообразовательных отношений, тогда как векторы fastText передают информацию как о близости лексических значений слов, так и об их внутренней форме (в понимании А. А. Потебни).



(а) (б)

Рис. 3. Векторизация целевых слов: а) в ruRoBERTa; б) в fastText.

ЗАКЛЮЧЕНИЕ

В ходе исследования была предпринята попытка оценить способность русскоязычных языковых моделей воспроизводить синтагматические связи между морфемами внутри производных слов и на уровне словообразовательных гнезд. Основное внимание было сосредоточено на проверке гипотезы о композициональности производных слов при агрегации векторов морфем.

Эксперименты с моделями fastText и ruRoBERTa-large показали, что наилучшие результаты могут быть получены с использованием усреднения для агрегации векторов морфем, при этом сравнение вектора основы с разностью вектора слова и агрегированного вектора морфем демонстрирует более высокие значения, чем сравнение вектора слова с агрегированным вектором морфем.

Эксперимент по оценке способности моделей воспроизводить семантические отношения внутри словообразовательных гнезд показал, что модель fastText лучше передает информацию как о близости лексических значений слов, так и об их внутренней форме, в то время как модель ruRoBERTa-large регистрирует семантическую близость векторов слов без учета их морфемного состава и словообразовательных отношений.

Гипотеза о композициональности производных слов при агрегации морфемных векторов не получила однозначного подтверждения. Как показал эксперимент с моделями fastText, наилучшие результаты агрегации векторов морфем

достигаются с использованием усреднения, при этом сравнение вектора основы с разностью вектора слова и агрегированного вектора морфем дает более высокие значения близости, чем сравнение вектора слова с агрегированным вектором морфем. При оценке семантических связей слов внутри словообразовательных гнезд модель fastText подтвердила способность учитывать как близость значений слов, так и их словообразовательные связи, тогда как модель ruRoBERTa воспроизводит преимущественно лексико-семантические отношения.

Перспективы развития настоящего исследования связаны с разработкой специализированных нейросетевых архитектур, учитывающих синтагматические отношения между морфемными сегментами внутри слов, совершенствованием морфемных токенизаторов, интегрируемых в языковые модели, расширением наборов данных для решения вышеуказанных задач, а также с развитием комбинированных подходов, объединяющих преимущества моделей семейств fastText и BERT.

СПИСОК ЛИТЕРАТУРЫ

1. *Гердд А.С.* Морфемика. СПб.: Изд-во С.-Петерб. ун-та, 2004. 176 с.

2. *Bolshakova E.I., Sapin A.S.* Building a Combined Morphological Model for Russian Word Forms. In: Burnaev E. et al. Analysis of Images, Social Networks and Texts. AIST 2021. Lecture Notes in Computer Science, vol. 13217. Springer, Cham, 2022. P. 45–55. https://doi.org/10.1007/978-3-031-16500-9_5

3. *Bolshakova E.I., Sapin A.S.* Building Dataset and Morpheme Segmentation Model for Russian Word Forms. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference «Dialogue». Moscow, 2021. P. 154–161. https://doi.org/10.28995/2075-7182-2021-20-154-161

4. *Morozov D., Shcherbakova O., Glazkova A.* Russian Neural Morpheme Segmentation: From Lemmata to Wordforms. In: Bakaev M. et al. Internet and Modern Society. IMS 2025. Communications in Computer and Information Science, vol. 2671. Springer, Cham, 2025. https://doi.org/10.1007/978-3-032-04958-2_12, P. 157–167.

5. *Morozov D., Astapenka L., Glazkova A., Garipov T., Lyashevskaya O.* BERT-like Models for Slavic Morpheme Segmentation. In: Che W., Nabende J., Shutova E., Pilehvar M.T. (Eds.) Proceedings of the Annual Meeting of the Association

for Computational Linguistics. Association for Computational Linguistics, 2025. P. 6795–6815. (Proceedings of the Annual Meeting of the Association for Computational Linguistics). https://doi.org/10.18653/v1/2025.acl-long.337

6. *Sorokin A., Kravtsova A.* Deep convolutional networks for supervised morpheme segmentation of Russian language. In: Ustalov D., Filchenkov A., Pivovarova L., Zizka J. (Eds.) Artificial Intelligence and Natural Language. P. 3–10. Springer, Cham, 2018. https://doi.org/10.1007/978-3-030-01204-5_1

7. *Selkirk E.* The syntax of words. Camb. (Mass), 1982. 136 p.

8. *Skalička V.* Hyposyntax. In: Slovo a slovesnost. Vol. 31. 1970. P. 1–6.

9. *Кубрякова Е.С.* Основы морфологического анализа. М., 1974. 320 с.

10. *Лопатин В.В.* Грамматическое описание славянских языков // Словообразование как объект грамматического описания. М., 1974.

11. *Lees R.* The Grammar of English nominalizations. The Hague, 1963.

12. *Marchand H.* The Categories and Types of Present-day English Word-Formation. Wiesbaden, 1960.

13. *Фивейская Е.А.* Словообразовательное моделирование семантики отглагольных имен в аспекте теории пропозиции // Сибирский филологический журнал. 2010 (3). С. 127–133.

14. *Филлмор Ч.* Дело о падеже // Новое в зарубежной лингвистике. Вып. 10. М., 1981.

15. *Шадрин В.И.* Семантика морфологических компонентов производных слов английского языка в свете категорий падежной грамматики // Морфемика. Принципы сегментации, отождествления и классификации морфологических единиц / Под ред. С.И. Богданова, А.С. Герда. СПб., 1997. С. 171–177.

16. *Morfessor.* URL: https://github.com/aalto-speech/morfessor, дата обращения 24.03.2026

17. *RussianMorphParsing.*
URL: https://github.com/alesapin/RussianMorphParsing, дата обращения 24.03.2026

18. *ruMorpheme*. URL: https://github.com/EvilFreelancer/ruMorpheme, дата обращения 24.03.2026

19. *Neuromodels.*
URL: https://ruscorpora.ru/license-content/neuromodels/, дата обращения

24.03.2026

20. *Asgari E., El Kheir Y., Sadraei Javaheri M. A.* MorphBPE: A Morpho-Aware Tokenizer Bridging Linguistic Complexity for Efficient LLM Training Across Morphologies, 2025. https://doi.org/10.48550/arXiv.2502.00894

21. *Teklehaymanot et al.* MoVoC: Morphology-Aware Subword Construction for Ge'ez Script Languages. In: Findings of the Association for Computational Linguistics: EMNLP 2025, p. 13131–13144, Suzhou, China. Association for Computational Linguistics, 2025. https://doi.org/10.48550/arXiv.2509.08812

22. *Nzeyimana A., Niyongabo Rubungo A.* KinyaBERT: a Morphology-aware Kinyarwanda Language Model. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), p. 5347–5363, Dublin, Ireland. Association for Computational Linguistics, 2022. https://doi.org/10.48550/arXiv.2203.08459

23. *Потиха З.А.* Школьный словарь строения слов русского языка: Пособие для учащихся. 2-е изд., испр. М.: Просвещение, 1999. 318 с.

24. *Тихонов А.Н.* Морфемно-орфографический словарь русского языка. М.: АСТ: Астрель, 2002. 704 с.

25. *Bojanowski P., Grave E., Joulin A., Mikolov T.* Enriching Word Vectors with Subword Information. In: Transactions of the Association for Computational Linguistics, 2017. P. 135-146. https://doi.org/10.48550/arXiv.2309.10931

26. *RusVectōrēs*. URL: https://rusvectores.org/ru/models/, дата обращения 24.03.2026

27. *Zmitrovich D., Abramov A., Kalmykov A., Kadulin V., Tikhonova M., Taktasheva E., Astafurov D., Baushenko M., Snegirev A., Shavrina T., Markov S., Mikhailov V., Fenogenova A.* A Family of Pretrained Transformer Language Models for Russian. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Torino, Italia, 2024. P. 507–524. https://doi.org/10.48550/arXiv.2309.10931

28. *ruRoBERTa-large.* URL: https://huggingface.co/ai-forever/ruRoBERTa-large, дата обращения 24.03.2026

REPRESENTATION OF INTRAWORD SYNTAGMATIC RELATIONS IN VECTOR LANGUAGE MODELS

D. K. Rodionova[1] [0009-0004-6296-8532], O. A. Mitrofanova[2] [0000-0002-3008-5514]

[1, 2]*Saint-Petersburg State University, Saint-Petersburg, Russia*

[1]*Chebyshev Research Center, Saint-Petersburg, Russia*

[1]rodionowadarja@yandex.ru, [2]o.mitrofanova@spbu.ru

Abstract

The paper discusses semantic structure representation of derivatives in language models, taking into account the intraword syntagmatic relations between derivational morphemes. Experiments were conducted using morphemic models developed by the Russian National Corpus (RNC), as well as fastText and ruRoBERTa models. The study is aimed at the verification of the hypothesis dealing with compositionality of derived words which are represented as aggregated morpheme vectors. In experiments we explore the representation of semantic relationships using fastText morpheme vectors and standard subword vectors in ruRoBERTa. The results indicate moderate sensitivity of fastText vectors to syntagmatic relations between morphemes as well as to derivational types. At the same time, it was found that aggregating morpheme vectors in fastText provides better representation of semantic relations between words compared to aggregating subword vectors in ruRoBERTa.

Standard BPE (Byte-Pair Encoding) and WordPiece tokenizers used in Transformer-based models are poorly interpretable with respect to linguistic data, as word segments do not always correspond to morphemes. The research problem lies in the need to assess the extent to which modern language models can capture linguistic features that characterize the relationships of derived words within word-formation families. The aim of the study is to evaluate the ability of predictive distributed vector embedding models to reproduce syntagmatic connections between morphemes within derived words and at the level of word-formation families in the Russian language.

The obtained results encourage the development of neural network architec-

tures that take into account syntagmatic relations between morphemes, the improvement of morpheme tokenizers, and their integration into language models.

***Keywords**: language models, morphemic analysis, word-formation methods, compositionality.*

REFERENCES

1. *Gerd A.S.* Morphology. St. Petersburg: Publishing House of St. Petersburg University, 2004. 176 p.

2. *Bolshakova E.I., Sapin A.S.* Building a Combined Morphological Model for Russian Word Forms. In: Burnaev, E., et al. Analysis of Images, Social Networks and Texts. AIST 2021. Lecture Notes in Computer Science, vol. 13217. Springer, Cham, 2022. P. 45–55. https://doi.org/10.1007/978-3-031-16500-9_5

3. *Bolshakova E.I., Sapin A.S.* Building Dataset and Morpheme Segmentation Model for Russian Word Forms. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference “Dialogue”. Moscow, 2021. P. 154–161. https://doi.org/10.28995/2075-7182-2021-20-154-161

4. *Morozov D., Shcherbakova O., Glazkova A.* Russian Neural Morpheme Segmentation: From Lemmata to Wordforms. In: Bakaev M. et al. Internet and Modern Society. IMS 2025. Communications in Computer and Information Science, vol. 2671. Springer, Cham, 2025. P. 157–167.
https://doi.org/10.1007/978-3-032-04958-2_12

5. *Morozov D., Astapenka L., Glazkova A., Garipov T., Lyashevskaya O.* BERT-like Models for Slavic Morpheme Segmentation. In: Che W., Nabende J., Shutova E., Pilehvar M.T. (Eds.) Proceedings of the Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2025. P. 6795–6815 (Proceedings of the Annual Meeting of the Association for Computational Linguistics). https://doi.org/10.18653/v1/2025.acl-long.337

6. *Sorokin A., Kravtsova A.* Deep convolutional networks for supervised morpheme segmentation of Russian language. In: Ustalov D., Filchenkov A., Pivovarova L., Zizka J. (Eds.) Artificial Intelligence and Natural Language. P. 3–10. Springer, Cham, 2018. https://doi.org/10.1007/978-3-030-01204-5_1

7. *Selkirk E.* The syntax of words. Camb. (Mass), 1982. 136 p.

8. *Skalička V.* Hyposyntax. In: Slovo a slovesnost. Vol. 31. 1970. P. 1–6.

9. *Kubryakova E.S.* Fundamentals of Morphological Analysis. Moscow, 1974. 320 p.

10. *Lopatin V.V.* Grammatical Description of Slavic Languages // Word Formation as an Object of Grammatical Description. Moscow, 1974.

11. *Lees R.* The Grammar of English nominalizations. The Hague, 1963.

12. *Marchand H.* The Categories and Types of Present-day English Word-Formation. Wiesbaden, 1960.

13. *Fiveyskaya E.A.* Word-Formation Modeling of the Semantics of Verbal Nouns in the Aspect of Proposition Theory // Siberian Philological Journal. 2010(3). P. 127–133.

14. *Fillmore C.* The Case for Case // New in Foreign Linguistics. Issue 10. Moscow, 1981.

15. *Shadrin V.I.* The Semantics of Morphological Components of Derived Words in the English Language in Light of the Categories of Case Grammar // Morphemics. Principles of Segmentation, Identification, and Classification of Morphological Units / Ed. by S.I. Bogdanov, A.S. Gerd. St. Petersburg, 1997. P. 171–177.

16. *Morfessor*. URL: https://github.com/aalto-speech/morfessor, last access 24.03.2026

17. *RussianMorphParsing.*
URL: https://github.com/alesapin/RussianMorphParsing, last access 24.03.2026

18. *ruMorpheme.* URL: https://github.com/EvilFreelancer/ruMorpheme, last access 24.03.2026

19. *Neuromodels.*
URL: https://ruscorpora.ru/license-content/neuromodels/, last access 24.03.2026

20. *Asgari E., El Kheir Y., Sadraei Javaheri M.A.* MorphBPE: A Morpho-Aware Tokenizer Bridging Linguistic Complexity for Efficient LLM Training Across Morphologies, 2025. https://doi.org/10.48550/arXiv.2502.00894

21. *Teklehaymanot et al.* MoVoC: Morphology-Aware Subword Construction for Ge'ez Script Languages. In: Findings of the Association for Computational Linguistics: EMNLP 2025, p. 13131–13144, Suzhou, China. Association for Computational Linguistics, 2025. https://doi.org/10.48550/arXiv.2509.08812

22. *Nzeyimana A., Niyongabo Rubungo A.* KinyaBERT: a Morphology-aware

Kinyarwanda Language Model. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), P. 5347–5363, Dublin, Ireland. Association for Computational Linguistics, 2022. https://doi.org/10.48550/arXiv.2203.08459

23. *Potikha Z.A.* School Dictionary of Word Structure of the Russian Language: A Guide for Students. 2nd ed., revised. Moscow: Prosveshchenie, 1999. 318 p.

24. *Tikhonov A.N.* Morphemic-Orthographic Dictionary of the Russian Language. Moscow: AST: Astrel, 2002. 704 p.

25. *.Bojanowski P., Grave E., Joulin A., Mikolov T.* Enriching Word Vectors with Subword Information. In: Transactions of the Association for Computational Linguistics, 2017. P. 135–146. https://doi.org/10.48550/arXiv.2309.10931

26. *RusVectōrēs.* URL: https://rusvectores.org/ru/models/, last access 24.03.2026

27. *Zmitrovich D., Abramov A., Kalmykov A., Kadulin V., Tikhonova M., Taktasheva E., Astafurov D., Baushenko M., Snegirev A., Shavrina T., Markov S., Mikhailov V., Fenogenova A.* A Family of Pretrained Transformer Language Models for Russian. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Torino, Italia, 2024. P. 507–524. https://doi.org/10.48550/arXiv.2309.10931

28. *ruRoBERTa-large.*
URL: https://huggingface.co/ai-forever/ruRoBERTa-large, last access 24.03.2026

СВЕДЕНИЯ ОБ АВТОРАХ

РОДИОНОВА Дарья Кирилловна – магистрант кафедры математической лингвистики филологического факультета Санкт-Петербургского государственного университета, старший инженер-программист Chebyshev Research Center. В 2014 году закончила бакалавриат на кафедре информационных систем в области искусств и гуманитарных наук факультета искусств Санкт-Петербургского государственного университета. В 2018 году окончила обучение Computer Science Center при поддержке компании JetBrains, слушала курсы ШАДа. Основные научные интересы связаны с языковым моделированием, математической лингвистикой, информационным поиском, извлечением знаний, анализом кода методами NLP и машинным обучением.

Daria Kirillovna RODIONOVA – master student at the Department of Mathematical Linguistics, Faculty of Philology, Saint-Petersburg State University, and senior software engineer at Chebyshev Research Center. In 2014 she graduated with a Bachelor degree from the Department of Information Systems in the Arts and Humanities at the Faculty of Arts of Saint-Petersburg State University. In 2018 she completed training at Computer Science Center supported by JetBrains and attending courses at the Yandex School of Data Analysis. Her main scientific interests are related to language modeling, mathematical linguistics, information retrieval, knowledge extraction, code analysis using NLP methods, and machine learning.

email: rodionowadarja@yandex.ru

ORCID: 0009-0004-6296-8532

МИТРОФАНОВА Ольга Александровна – кандидат филологических наук, доцент кафедры математической лингвистики филологического факультета Санкт-Петербургского государственного университета. В 1995 году закончила отделение математической лингвистики филологического факультета Санкт-Петербургского государственного университета, в 1999 году защитила диссертацию на соискание ученой степени кандидата филологических наук по специальности 10.02.21 – Прикладная и математическая лингвистика. Является автором более 150 публикаций в области компьютерной и корпусной лингвистики. Основные научные интересы связаны с моделями языка, машинным обучением, автоматическим пониманием и генерацией текстов, лингвистикой конструкций, дистрибутивной семантикой, тематическим моделированием.

Olga Aleksandrovna MITROFANOVA – PhD in Philology, associate professor at the Department of Mathematical Linguistics, Faculty of Philology, Saint-Petersburg State University. She graduated from Mathematical Linguistics Department, Faculty of Philology, Saint-Petersburg State University in 1995, and in 1999 she defended her thesis in Applied and Mathematical Linguistics (10.02.21). She is the author of over 150 publications in the field of Computational and Corpus Linguistics. Her main research interests are language models, machine learning, natural text understanding and generation, construction linguistics, distributional semantics, and topic modeling.

email: o.mitrofanova@spbu.ru

ORCID: 0000-0002-3008-5514

Материал поступил в редакцию 23 марта 2026 года

УДК 519.252+519.254+004.75

МЕТОДЫ АВТОМАТИЗИРОВАННОГО ИЗВЛЕЧЕНИЯ ПАРАМЕТРОВ И ОПИСАНИЙ ПРОГРАММ ДЛЯ ИНТЕГРАЦИИ ИХ НА ВЫЧИСЛИТЕЛЬНЫЕ КОМПЛЕКСЫ

Т. В. Санников[1] [0009-0004-1144-8836], А. Н. Сальников[2] [0000-0001-8669-9905]

[1, 2]*Московский государственный университет им. М. В. Ломоносова, г. Москва, Россия*

[2]*Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Россия*

[1]timohaj1@yandex.ru, [2]salnikov@cs.msu.ru

Аннотация

Рассмотрена проблема координации разнородных программных средств в гетерогенных средах распределенного запуска приложений. Ручное конфигурирование параметров запуска для вновь устанавливаемых программ на вычислительный кластер (таких как ключи командной строки, значения переменных окружения и настройки конфигурационных файлов) создает серьезные трудности для исследователей предметных областей из-за больших объемов служебной информации и необходимости сохранения и агрегации информации в некотором фиксированном формате. Предложен метод автоматизированного извлечения параметров запуска, базирующийся на гибридной архитектуре обучения нейронной сети, сочетающей генерацию обучающей выборки большими языковыми моделями и последующее дообучение компактного трансформерного энкодера. Реализация подхода исключает зависимость от дорогостоящих графических ускорителей за счет применения методики низкоранговой адаптации (Low-Rank Adaptation) для моделей размером до 1 млрд параметров, что обеспечивает возможность выполнения модели (инференса) на обычных центральных процессорах управляющих узлов. Для формализации качества извлечения разработана двухкомпонентная метрика, агрегирующая структурную корректность выходной JSON-схемы (наличие в полученных дан-

ных обязательных полей, типов параметров программы) и семантическую точность значений параметров (соответствие описания в документации). Экспериментальная оценка метода ориентирована на корпус документации программных пакетов (man-страницы, README). Результаты проектирования подтверждают возможность аппроксимации процесса анализа документации компактной моделью, что способствует автоматизации жизненного цикла развертывания программного обеспечения и снижению ошибок управления потоками задач в распределенных вычислительных комплексах.

Ключевые слова: *низкоранговая адаптация, извлечение данных, анализ программного кода, автоматизация запуска, обработка естественного языка, научная рабочая среда, высокопроизводительные вычисления.*

ВВЕДЕНИЕ

Распространение высокопроизводительных вычислительных ресурсов в современной научной среде претерпевает существенные изменения. Если ранее доступ к мощным вычислительным комплексам был привилегией узкого круга специализированных организаций, то в настоящее время наблюдается тенденция к демократизации доступа. Исследовательские группы все чаще получают возможность использовать несколько вычислительных комплексов одновременно для решения своих задач, что позволяет из этих комплексов составлять распределенную систему запуска приложений [1].

Существующая практика настройки программного обеспечения в подобных системах зачастую опирается на ручное вмешательство специалиста. Пользователь вынужден самостоятельно анализировать сопроводительную документацию, изучать исходный текст программ и вручную задавать параметры запуска. Такой подход имеет ряд недостатков. Во-первых, он требует от исследователя глубоких знаний в области системного программирования и архитектуры вычислительных комплексов, что не всегда соответствует профилю специалиста предметной области. Биолог, физик или химик может быть экспертом в своей науке, но не обладать достаточной квалификацией для тонкой настройки программного окружения. Во-вторых, человеческий фактор неизбежно ведет к ошибкам и неполному заполнению конфигурационных данных. Разнообразие

способов задания параметров в различном программном обеспечении, неполнота и неоднозначность документации, а также различия в форматах похожих параметров усугубляют ситуацию [2].

Цель настоящей работы заключается в облегчении пользователю процесса интеграции нового программного обеспечения в вычислительный кластер за счет разработки некоторого программного инструмента. Этот инструмент, просмотрев» исходные коды программного обеспечения и документацию, автоматически построит в некотором формате строгое формальное описание параметров программ для автоматизации дальнейшего запуска данных программ внешними средствами запуска на кластере. Программный инструмент также составит «объяснение» назначения каждого параметра в строгом формальном виде для автоматизации интеграции с внешними программами и сайтами.

Создание такого программного инструмента в прошлом было затруднено из-за большого разнообразия форм описания параметров в исходных кодах и документации. По сути применялись методы статического анализа кода и некоторые приемы в духе запустить много разнообразных grep (команды поиска и фильтрации по шаблону). Однако сейчас эта сложность частично преодолена за счет развития графических процессоров и больших нейросетевых моделей, натренированных на гигантских объемах программного кода, что делает возможным создание инструмента по автоматическому «узнаванию» параметров и описаний в коде.

1. СИСТЕМА РАСПРЕДЕЛЕННОГО ЗАПУСКА ПРИЛОЖЕНИЙ

Система распределенного запуска приложений [3] представляет собой программный комплекс, предназначенный для координации выполнения вычислительных задач на нескольких вычислительных кластерах. Архитектура системы построена по клиент-серверному принципу, где центральный узел принимает запросы от исследовательских групп и осуществляет диспетчеризацию задач по доступным вычислительным ресурсам. В основе системы лежит сервер распределения, который выполняет функции координационного центра. Сервер получает от пользователей описания потоков задач с зависимостями, анализирует доступные ресурсы кластеров и формирует оптимальное расписание выполнения задач. Ключевой особенностью архитектуры является отделе-

ние логики предметной области от инфраструктуры исполнения, что позволяет адаптировать систему к различным вычислительным окружениям без модификации исходного кода научных программ.

Принципиальным компонентом системы является механизм управления зависимостями между задачами. Зависимости определяются через файлы данных, которые передаются от выполненных задач к зависимым от них (рис. 1).

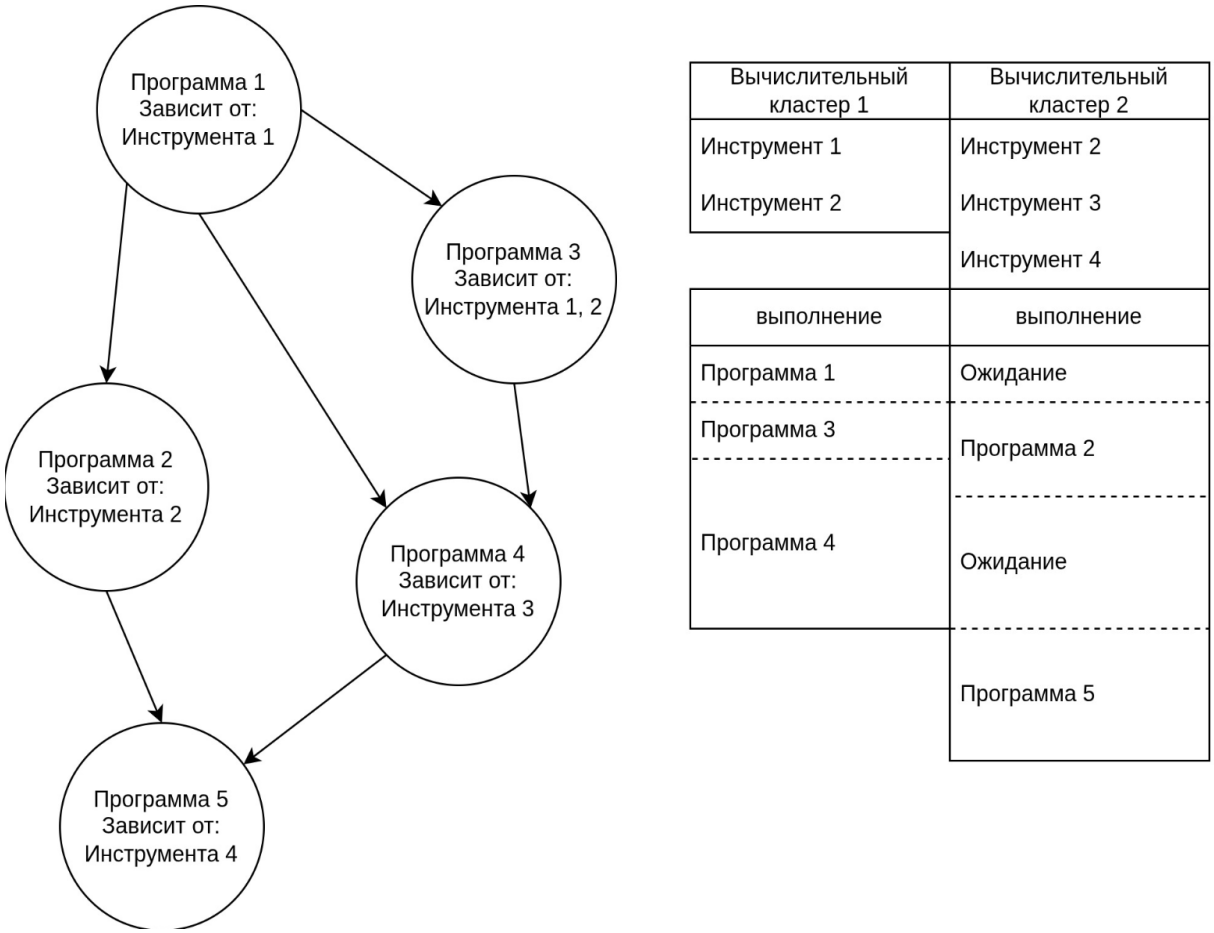


Рис. 1. Иллюстрация потока задач и его распределения на графе.

Система отслеживает состояние выполнения каждой задачи и обеспечивает передачу необходимых файлов только после успешного завершения решения предшествующих задач. Как правило, содержательные параметры научных программ, входные и выходные данные представлены именно фай-

лами и по соответствующим типам файлов образуют группу по различным вариантам обработки данных. Пользователь формирует описание вычислительного потока, которое преобразуется в структурированный формат. Генератор конфигурационных данных извлекает параметры программного обеспечения и создает машиночитаемое описание в формате JSON. Программа распределения задач использует полученные данные для формирования оптимального расписания и размещения задач по кластерам.

2. ОБЗОР МЕТОДОВ ИЗВЛЕЧЕНИЯ ПАРАМЕТРОВ

2.1. Критерии обзора

Для проведения систематического обзора инструментов анализа программного обеспечения был сформирован набор критериев, описывающих ключевые требования к системе автоматизированного развертывания в распределенной вычислительной среде. Выбор таких критериев обусловлен необходимостью интеграции инструмента в существующий конвейер оркестрации задач и минимизации ручного вмешательства на этапе конфигурирования.

Первостепенное значение имеет уровень автоматизации процесса извлечения параметров. Способность инструмента работать без предварительной ручной разметки кода или документации определяет возможность его применения для массового развертывания программного обеспечения на кластерах. Инструменты, требующие декларативного описания параметров непосредственно в исходном коде, накладывают ограничения на использование сторонних программных пакетов, модификация которых нежелательна или невозможна.

Формат вывода данных выступает вторым критическим критерием. Наличие структурированного представления результатов в машиночитаемом формате, таком как JSON, или формате, который возможно преобразовать в него, является обязательным условием для последующей обработки системой управления задачами.

Источники данных, анализируемые инструментом, определяют полноту извлекаемой информации. Поддержка анализа сопроводительной документации, man-страниц и исходного кода одновременно позволяет охватить различ-

ные способы описания параметров в программных продуктах. Документация часто содержит сведения о параметрах, которые не очевидны из статического анализа кода, включая ограничения на значения и семантические описания.

Возможность программного вызова через интерфейс командной строки или программный интерфейс критична для встраивания инструмента в автоматизированный конвейер развертывания. Инструменты, предназначенные исключительно для интерактивного использования, не могут быть интегрированы в процессы непрерывной интеграции и автоматического тестирования.

2.2. Сравнение существующих методов

Инструменты CarpetFuzz [4] и FuzzGen [5] разработаны для решения задач безопасности и тестирования программного обеспечения, а не для автоматизации развертывания в распределенных вычислительных средах. В табл. 1 представлены результаты сравнения инструментов извлечения параметров. (зеленый положиельная характеристика, красный отрицательная)

Табл. 1. Сравнительный анализ инструментов извлечения параметров.

Инструмент	Автоматизация	JSON	Документация	API/CLI	Разметка	Исходный код	Ограничения
CarpetFuzz	+	-	+	+	-	-	Для тестирования на уязвимости
FuzzGen	+	-	-	+	-	+	Требует исходный код
ArgParse	-	+	-	+	+	+	Ручная декларация
Click/Typer	-	+	-	+	+	+	Фремворки
Sphinx/ Doxygen	-	+	+	-	+	+	Генерация документации

CarpetFuzz специализируется на фаззинг-тестировании путем сопоставления документации с реализацией функций, что позволяет выявлять уязвимости и несоответствия в обработке аргументов. FuzzGen ориентирован на генерацию тестовых оберток для программ с использованием анализа исходного кода. Оба инструмента обеспечивают высокий уровень автоматизации и не требуют ручной разметки, однако их архитектура не предусматривает формирования структурированных данных о параметрах запуска, пригодных для последующей обработки системой оркестрации задач.

Разработка собственного метода извлечения параметров обусловлена необходимостью получения машиночитаемых конфигурационных данных в формате JSON для интеграции с системой распределенного запуска. Существующие решения либо требуют ручной декларации параметров в коде, либо не предоставляют программный интерфейс для автоматизации, либо ориентированы на генерацию документации вместо конфигурационных файлов. Предлагаемый подход сочетает автоматизацию анализа документации через языковые модели с возможностью программного вызова и выводом структурированных данных, что позволяет устранить выявленные недостатки и обеспечить автоматизацию процесса размещения программного обеспечения в гетерогенной вычислительной среде.

3. МЕТОД АВТОМАТИЗАЦИИ ИЗВЛЕЧЕНИЯ ПАРАМЕТРОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Предложенный подход основан на использовании языковых моделей большого размера для генерации обучающих данных и последующем дообучении компактной трансформерной архитектуры, способной функционировать без специализированных графических ускорителей. Подобная двухэтапная стратегия позволяет сочетать высокую точность извлечения параметров с практической применимостью в реальных вычислительных средах, где доступ к GPU-ресурсам может быть ограничен или экономически нецелесообразен. Разработанный метод состоит из четырех последовательных этапов, представленных на рис. 2, каждый из которых решает определенную задачу в процессе создания инструмента автоматизации.

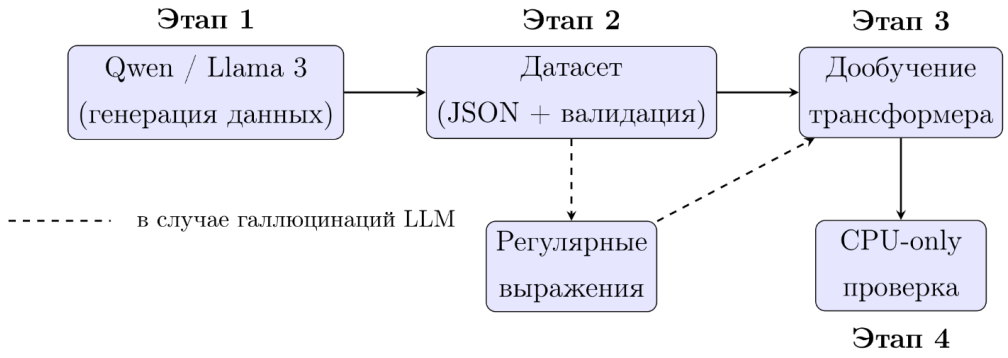


Рис. 2. Этапы реализации инструментов автоматизации на основе 4 этапов.

3.1. Этап формирования «сырого» датасета

Первоначальный этап предполагает сбор и систематизацию документации программного обеспечения, предназначенного для анализа. Источниками данных выступают сопроводительные материалы программных пакетов, включая man-страницы, файлы README, документацию в форматах reStructuredText и Markdown, а также комментарии, встроенные в исходном коде. Выбор разнообразных источников обусловлен необходимостью охвата различных способов описания параметров в программных продуктах.

Формирование «сырого» датасета включает следующие процедуры:

1) автоматизированный сбор документации из репозиториев программного обеспечения;

2) нормализация текстовых данных (удаление форматирования, унификация кодировок);

3) сегментация документов на логические блоки, соответствующие описаниям отдельных параметров;

4) метаданные о источнике документа, версии программного обеспечения и дате публикации [6].

Объем «сырого» датасета определяется количеством программных пакетов, подлежащих анализу в рамках системы распределенного запуска. Для обеспечения репрезентативности выборки рекомендуется включать программы с различными способами задания параметров: флаги командной строки, конфигурационные файлы, переменные окружения. Минимальный рекомен-

дуемый объем составляет 500–1000 образцов документации, что обеспечивает достаточное разнообразие лингвистических конструкций для последующего обучения модели.

3.2. Этап разметки данных языковой моделью

Второй этап является ключевым компоненом метода, где осуществляется автоматизированная генерация структурированных описаний параметров на основе текстов документации. Для решения этой задачи используются предварительно обученные языковые модели большого размера, такие как Qwen или Llama 3. Выбор данных архитектур обусловлен их способностью к пониманию контекста и извлечению семантических связей из неструктурированного текста.

Процесс разметки включает следующие шаги:

1) формирование промптов для языковой модели, содержащих инструкцию по извлечению параметров и пример желаемого формата вывода;

2) последовательная обработка документов сырого датасета через API языковой модели;

3) парсинг полученных ответов и приведение их к единой схеме представления данных;

4) проверка правильности структуры JSON-файла;

5) первичная автоматическая валидация с использованием запуска программ в изолированной среде.

Выходные данные этапа представляют собой JSON-объекты, содержащие для каждого параметра следующие поля: имя параметра, тип значения, значение по умолчанию, текстовое описание и ограничения на допустимые значения. Подобная структура соответствует требованиям системы оркестрации задач и позволяет непосредственно использовать извлеченные данные для генерации конфигурационных файлов.

Важным аспектом данного этапа является управление качеством генерации. Языковые модели склонны к галлюцинациям – формированию правдоподобных, но фактически неверных утверждений. Для минимизации этого риска применяются стратегия множественной генерации с последующим согласованием результатов, а также включение в промпт явных указаний на необходимость описания источника информации в тексте документации. Кроме того, ис-

пользуется подход с попыткой запуска программы по сформированному json и немедленному ее прекращению для проверки минимальной работоспособности.

3.3. Этап ручной валидации датасета

Третий этап предполагает экспертную проверку сгенерированных разметок для обеспечения достоверности обучающих данных. Ручная валидация осуществляется специалистами, обладающими знаниями в области системного программирования и анализа программного обеспечения.

Процедура валидации включает:

- сверку извлеченных параметров с исходным текстом документации;
- проверку корректности типов значений (целочисленные, строковые, булевы, пути к файлам);
- верификацию значений по умолчанию путем сопоставления с примерами использования в документации;
- контроль полноты описания ограничений на значения параметров.

Для формализации процесса валидации разработан чек-лист, содержащий критерии оценки качества каждой разметки. Каждый параметр оценивается по пятикомпонентной метрике полноты описания, представленной в следующем подразделе. Параметры, не прошедшие валидацию, возвращаются на этап разметки для повторной обработки скорректированными промптами. Результатом этапа становится верифицированный датасет, пригодный для использования в качестве обучающей выборки при дообучении трансформерной модели.

3.4. Этап дообучения трансформерной модели

Завершающий этап метода предполагает дообучение компактной трансформерной архитектуры на верифицированном датасете. Выбор легкой модели (не более 1 млрд параметров) обусловлен требованием возможности развертывания программы на центральных процессорах без использования графических ускорителей. Подобное ограничение критично для развертывания инструмента непосредственно на управляющих узлах системы распределенного запуска, где выделение специализированных вычислительных ресурсов

нецелесообразно, тем более, что сервера исследовательских групп могут не иметь GPU, что делает проект бесполезным для такого типа серверов.

Для дообучения применяется методика LoRA (Low-Rank Adaptation) [7], позволяющая эффективно адаптировать предварительно обученную модель к конкретной задаче при минимальных вычислительных затратах. Суть метода заключается в добавлении низкоранговых матриц к слоям внимания исходной модели, при этом основные веса модели остаются замороженными. Подобный подход снижает объем требуемой памяти и ускоряет процесс обучения.

Конфигурация дообучения включает следующие параметры:

- базовая архитектура: трансформер с 6–8 слоями внимания;
- размерность скрытого представления: 512–768;
- ранг матриц LoRA: 8–16;
- размер пакета: 16–32 образца;
- количество эпох: 3–5 с ранней остановкой по валидационной метрике.

Валидационная выборка формируется из части верифицированного датасета и используется для контроля переобучения. Обучение прекращается при достижении плато на валидационной метрике или при превышении максимального количества эпох.

3.5. Регулярные выражения при галлюцинациях нейросетевой модели

Для решения проблем, когда нейросеть добавляет лишние слова, но при этом выдает json-файл как часть сгенерированных слов, применяются регулярные выражения, позволяющие вычленить смысловую часть из варианта ответа, предложенного нейросетью.

4. РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННОГО МЕТОДА

4.1. Метрики оценки качества

Для количественной оценки качества извлечения параметров разработана двухкомпонентная метрика, учитывающая как структурную корректность выходных данных, так и их семантическую точность.

Интегральная метрика качества вычисляется по формуле

$$Q = \alpha \cdot S_{struct} + (1 - \alpha) \cdot S_{sem},$$

где α = 0.6 — весовой коэффициент структурной корректности, S_{struct} — оценка структурного соответствия, S_{sem} — оценка семантической точности.

Структурная корректность S_{struct} оценивает соответствие выходных данных заданной схеме JSON и включает: 1) наличие всех обязательных ключей в объекте параметра; 2) соответствие типов значений, объявленным в схеме; 3) корректность вложенности структур данных.

Семантическая точность S_{sem} отражает содержательную правильность извлеченной информации: 1) соответствие типов значений фактическому назначению параметров; 2) точность значений по умолчанию относительно документации; 3) полнота описания ограничений на значения.

Дополнительно вводится индекс полноты описания параметров

$$C = \frac{\sum w_k \cdot I_k}{\sum w_k},$$

где $I_k \in \{0, 1\}$ — наличие k-го поля в извлеченном описании, w_k — весовой коэффициент поля.

Компоненты индекса полноты:

- I_1: имя параметра (w_1 = 1.0);
- I_2: тип значения (w_2 = 0.8);
- I_3: значение по умолчанию (w_3 = 0.7);
- I_4: текстовое описание (w_4 = 0.5);
- I_5: ограничения на значения (w_5 = 0.9).

Подобная система метрик позволяет проводить детальный анализ качества работы модели на различных аспектах задачи и выявлять направления для улучшения метода.

4.2. Результаты тестирования

В ходе экспериментального исследования была проведена оценка эффективности извлечения параметров командной строки с использованием языковой модели Qwen2.5-7B, базовой предобученной версии трансформера Qwen2.5-0.5B (Base), специализированного адаптера (LoRA), дообученного на данных man-страниц, и исходной модели (Qwen2.5-0.5B). Полученные результаты демонстрируют кардинальное превосходство адаптированного решения:

метрика точности (accuracy) для конфигурации LoRA достигла значения 0.867 ± 0.340, что более чем в три раза превышает показатели базовой модели Qwen2.5-0.5B (0.267 ± 0.442) и исходного варианта Qwen2.5-7B, показавшего низкий результат 0.641 ± 0.321. Аналогичная динамика наблюдается в показателях полноты (completeness), где метод LoRA обеспечил значение 0.715 ± 0.290 против 0.236 ± 0.210 у базы и критически низких 0.140 ± 0.246 у необученной модели, что свидетельствует о неспособности архитектуры «из коробки» корректно интерпретировать жесткую структуру технической документации без предварительной настройки весов.

Низкие показатели исходной модели Qwen2.5-7B подтверждают гипотезу о том, что универсальные языковые модели, не прошедшие адаптацию на репрезентативной выборке специфических форматов описания аргументов, склонны к ошибкам сегментации и не могут надежно выделять семантические связи между опциями и их параметрами в контексте системной документации.

Таким образом, можно заключить, что применение техники эффективной донастройки (LoRA) на специализированном корпусе man-страниц является критически необходимым условием для создания работоспособной системы автоматизированного развертывания программного обеспечения в гетерогенных вычислительных средах. Достигнутый уровень точности 0.867 и полноты 0.715 подтверждает пригодность разработанного подхода для генерации машиночитаемых конфигурационных файлов в формате JSON. В то же время выявленная недостаточность обобщающей способности модели на текущем этапе диктует необходимость расширения обучающей выборки за счет других типов документации и исходного кода, что составит основное содержание дальнейших исследований, направленных на создание универсального инструмента, способного агрегировать параметры из любых доступных источников проекта без потери качества структурирования данных.

ЗАКЛЮЧЕНИЕ

За счет применения методики LoRA упрощена задача процесса конфигурирования программного обеспечения для распределенных вычислительных сред. Проведенный систематический обзор существующих инструментов статического анализа и генерации документации (CarpetFuzz, FuzzGen, Sphinx

и др.) позволил сформулировать набор критических требований, которым должны удовлетворять решения по автоматизации. Требованиям полностью не удовлетворяет ни одно из решений, рассмотренных в обзоре. На основе выявленных ограничений был разработан метод автоматического извлечения параметров запуска с использованием больших языковых моделей. Ключевым результатом стали формирование специализированного датасета на основе man-страниц и успешная реализация прототипа системы дообучения трансформерных архитектур (Qwen, Llama 3). Экспериментально подтверждено, что адаптация модели под предметную область технической документации позволяет достичь метрики точности (accuracy) на уровне 0.867, что делает предложенный подход пригодным для практического применения при помещении нового кода в вычислительные кластерные системы. Реализованный метод обеспечивает работу на центральных процессорах управляющих узлов без необходимости использования графических ускорителей.

Однако предложенная реализация метода имеет ряд ограничений: архитектура системы пока требует участия человека на этапе валидации сгенерированных разметок, что замедляет процесс формирования обучающих выборок для новых пакетов. В связи с этим дальнейшая работа будет сосредоточена на расширении источников данных и повышении степени автономности.

СПИСОК ЛИТЕРАТУРЫ

1. *Suter F. et al*. A terminology for scientific workflow systems // Future Generation Computer Systems. 2026. Vol. 174. P. 107974. https://doi.org/10.1016/j.future.2025.107974

2. *da Silva R.F. et al.* Workflows Community Summit 2024: Future Trends and Challenges in Scientific Workflows: tech. rep. ORNL/TM-2024/3573. Oak Ridge: Oak Ridge National Laboratory, 2024.

3. *Санников Т.В., Сальников А.Н.* Обработка потока задач с зависимостями на нескольких вычислительных кластерах // Параллельные вычислительные технологии XIX Всероссийская конференция с международным участием (ПаВТ'2025). Челябинск: Изд-во ЮУрГУ, 2025. С. 270–283. https://doi.org/10.14529/pct2025.

4. *Wang D., Li Y., Zhang Z., Chen K.* CarpetFuzz: Automatic Program Option

Constraint Extraction from Documentation for Fuzzing // Proc. of the 32nd USENIX Security Symposium. Anaheim: USENIX Association, 2023. P. 2847–2864.

5. *Ispoglou K., Austin D., Mohan V., Payer M.* FuzzGen: Automatic Fuzzer Generation // Proc. of the 29th USENIX Security Symposium (USENIX Security 20). Boston: USENIX Association, 2020. P. 1001–1018.

6. *Wilkinson S.R. et al.* Applying the FAIR principles to computational workflows // Scientific Data. 2025. Vol. 12, No. 1. Art. 328. https://doi.org/10.1038/s41597-025-04451-9

7. *Hu E.J. et al.* LoRA: Low-Rank Adaptation of Large Language Models // Proc. of the International Conference on Learning Representations (ICLR). 2022. 16 p.

URL: https://openreview.net/forum?id=nZeVKeeFYf9 (дата обращения: 28.03.2026).

METHODS FOR THE AUTOMATED EXTRACTION OF PROGRAM PARAMETERS AND DESCRIPTIONS FOR THEIR INTEGRATION INTO COMPUTING SYSTEMS

T. V. Sannikov[1] [0009-0004-1144-8836], A. N. Salnikov[2] [0000-0001-8669-9905]

[1, 2]*Lomonosov Moscow State University, Moscow, Russia*

[2]*Federal Research Center for Information Technologies, Russian Academy of Sciences, Moscow, Russia*

[1]timohaj1@yandex.ru, [2]salnikov@cs.msu.ru

Abstract

This article addresses the problem of coordinating heterogeneous software tools in heterogeneous distributed application execution environments. Here, manually configuring launch parameters for newly installed programs on a computing cluster (such as command-line switches, environment variable values, and configuration file settings) poses significant challenges for domain researchers due to the large volume of utility information and the need to store and aggregate information in a fixed format. We propose a method for the automated extraction of launch parameters based on a hybrid neural network training architecture that combines the

generation of training samples using large language models with the subsequent fine-tuning of a compact transformer encoder. This approach eliminates the need for expensive graphics accelerators by applying the Low-Rank Adaptation (LoRA) technique to models with up to 1 billion parameters, enabling model execution (inference) on standard CPUs in control nodes. To formalize the quality of extraction, a two-component metric has been developed that aggregates the structural correctness of the output JSON schema (the presence of required fields and program parameter types in the obtained data) and the semantic accuracy of parameter values (correspondence with the description in the documentation). The experimental evaluation of the method focuses on a corpus of software package documentation (man pages, README files). The design results confirm the possibility of approximating the documentation analysis process with a compact model, which contributes to the automation of the software deployment lifecycle and the reduction of task flow management errors in distributed computing systems.

Keywords: *low-rank adaptation (LoRA), data extraction, source code analysis, launch automation, natural language processing (NLP), scientific workflow, high-performance computing (HPC).*

REFERENCES

1. Suter F. et al. A terminology for scientific workflow systems // Future Generation Computer Systems. 2026. Vol. 174. 107974. https://doi.org/10.1016/j.future.2025.107974

2. *da Silva R.F. et al.* Workflows Community Summit 2024: Future Trends and Challenges in Scientific Workflows: tech. rep. ORNL/TM-2024/3573. Oak Ridge: Oak Ridge National Laboratory, 2024.

3. *Sannikov T.V., Salnikov A.N.* Processing of Task Streams with Dependencies on Multiple Computing Clusters // Parallel Computational Technologies – 19th International Conference on Parallel Computing Technologies (PaVT'2025): short papers and poster descriptions. Chelyabinsk: South Ural State University Publishing House, 2025. P. 270–283. https://doi.org/10.14529/pct2025.

4. *Wang D., Li Y., Zhang Z., Chen K.* CarpetFuzz: Automatic Program Option Constraint Extraction from Documentation for Fuzzing // Proc. of the 32nd USENIX

Security Symposium. Anaheim: USENIX Association, 2023. P. 2847–2864.

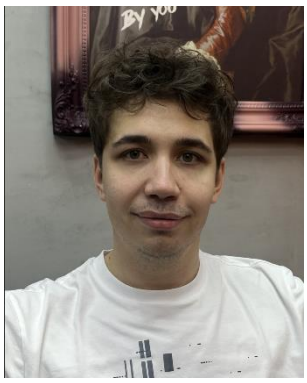
5. *Ispoglou K., Austin D., Mohan V., Payer M.* FuzzGen: Automatic Fuzzer Generation // Proc. of the 29th USENIX Security Symposium (USENIX Security 20). Boston: USENIX Association, 2020. P. 1001–1018.

6. *Wilkinson S.R. et al.* Applying the FAIR principles to computational workflows // Scientific Data. 2025. Vol. 12, No. 1. Art. 328.
https://doi.org/10.1038/s41597-025-04451-9

7. *Hu E. J. et al.* LoRA: Low-Rank Adaptation of Large Language Models // Proc. of the International Conference on Learning Representations (ICLR). 2022. 16 p.
URL: https://openreview.net/forum?id=nZeVKeeFYf9 (accessed 28.03.2026).

СВЕДЕНИЯ ОБ АВТОРАХ

.



САННИКОВ Тимофей Владимирович -- студент магистратуры Московского государственного университета имени Ломоносова. Область научных интересов: архитектура языковых моделей, высокопроизводительные системы.

Timofei Vladimirovich SANNIKOV -- a first-year master's student at Lomonosov Moscow State University.Research interests: language model architecture, scientific workflows, and high-performance systems.

email: timohaj1@yandex.ru;

ORCID: 0009-0004-1144-8836

САЛЬНИКОВ Алексей Николаевич – кандидат физико-математических наук, доцент кафедры Автоматизации систем вычислительных комплексов Московского государственного университета имени М.В. Ломоносова.

Научные интересы: параллельные и распределенные вычисления, интеллектуальный анализ данных, анализ сетей вычислительных кластеров.

Alexey Nikolaevich SALNIKOV – Philosophy doctor of Physics and Mathematics, Associate Professor, Lomonosov Moscow State University, department of Computer Science.

Research interests: parallel and distributed computations, data mining, computer cluster interconnections analysis.

email: salnikov@cs.msu.ru;

ORCID: 0000-0001-8669-9905

Материал поступил в редакцию 16 марта 2026 года

УДК 004.4

СИСТЕМА АВТОМАТИЧЕСКОГО ФОРМИРОВАНИЯ, ОБРАБОТКИ И УПРАВЛЕНИЯ МЕТАДАННЫМИ ДОКУМЕНТОВ ЭЛЕКТРОННЫХ КОЛЛЕКЦИЙ

А. Р. Хамеджанов[0009-0000-5624-2453]

Казанский (Приволжский) федеральный университет, г. Казань, Россия

hamedzhanovalmaz@gmail.com

Аннотация

В настоящее время издательский цикл претерпевает значительные технологические изменения: внедряются автоматизированные системы управления публикационными процессами, используются нейросетевые технологии для обработки контента, активно развиваются инструменты интеллектуального анализа научных данных. Одним из ключевых трендов становится автоматизация издательского цикла, направленная на ускорение обработки рукописей, повышение качества метаописания и обеспечение совместимости информационных ресурсов. В этом контексте метаданные выступают связующим элементом для машинной обработки и навигации в пространстве научных знаний, обеспечивая структурирование информации, ее интерпретацию и интеграцию в цифровые библиотечные системы. Однако метаданные научных публикаций часто содержат ошибки, неточности или являются неполными, а их ручное формирование и уточнение требуют значительных временных затрат и не обеспечивают высокой точности. В работе представлена система автоматического формирования, обработки и управления метаданными научных документов на основе данных, полученных из сервисов поиска научных публикаций и открытых баз знаний. Эта система может использоваться для автоматизации процесса извлечения, уточнения и дополнения метаданных научных публикаций с целью последующего формирования электронных коллекций научных документов.

***Ключевые слова**: цифровая математическая библиотека, семантическая сеть, автоматизация редакционных процессов, формирование метаданных, извлечение метаданных, дополнение метаданных, NISO JATS, цифровая библиотека.*

ВВЕДЕНИЕ

Метаданные являются не просто описанием данных, но и выступают связующим звеном, обеспечивающим структурирование знаний, его интерпретацию и возможность навигации в пространстве научных знаний, включая традиционные библиотечные каталоги или современные семантические веб-системы [1]. В компьютерных системах метаданные выполняют ключевую функцию обеспечения совместимости различных информационных ресурсов и автоматизированной обработки массивов данных [2]. В контексте цифровых библиотек качество метаданных напрямую определяет эффективность поиска, доступность контента и долгосрочную сохранность научного наследия [3]. Помимо автоматизированных систем, работающих с обработкой данных, метаданные нужны также для создания пользователями запросов, анализа данных и интерпретации их содержимого. В ряде источников также отмечается ключевая роль метаданных в обеспечении технических стандартов и правил генерации записей, что значительно упрощает процесс работы с данными [4, 5].

Однако метаданные по разным причинам могут содержать ошибки и неточности (например, в случаях, когда авторы имеют одно и то же полное имя). Кроме того, ручное формирование блока метаданных требует существенных временных затрат.

В рамках проведенного исследования для решения ряда отмеченных проблем, реализована система автоматического формирования, обработки и уточнения исходных метаданных и дополнения недостающей информации на основании полученных данных с помощью поисковой системы Google Scholar (https://scholar.google.com/), систем ORCID (https://orcid.org/), Yandex Translate (https://translate.yandex.ru/) и запросов к графу знаний WikiData (https://www.wikidata.org). Разработанная система позволяет извлекать метаданные на основе научных публикаций, загруженных в нее, и формировать выходной XML-файл в формате NISO JATS V1.0 (Journal Article Tag Suite, https://jats.nlm.nih.gov/1.0). Данные, которые не были указаны в статье, например ключевые слова или элементы аффилиации, могут быть дополнены из открытых источников.

Для эффективной интеграции сервисов в функционал цифровых библиотек, а также для обеспечения их совместимости с внешними библиотечными системами и базами данных критически важно уделять внимание согласованности форматов метаданных, используемых в этих информационных ресурсах. Цифровые математические библиотеки DML-CZ (Czech Digital Mathematics Library, https://dml.cz), Numdam (http://www.numdam.org) и EuDML (The European Digital Mathematics Library, https://initiative.eudml.org) используют XML-схемы NISO JATS V1.0 для описания публикаций из математических изданий согласно международным стандартам, предложенным в проекте Всемирной цифровой математической библиотеки (World Digital Mathematical Library – WDML) [6]. Поэтому коллекции метаданных, собранные для цифровой библиотеки Lobachevskii-DML, также должны соответствовать международным стандартам, чтобы обеспечить интеграцию этих коллекций в агрегирующие научные библиотеки.

Стандарт NISO JATS версии 1.0 представляет собой набор тегов для структурирования и описания научных статей в формате XML. Этот стандарт включает в себя множество полей, которые обеспечивают детализированное описание статьи. Информацию о конкретных тегах можно найти, например, в https://jats.nlm.nih.gov/1.0. Фрагмент XML-кода, приведенный на рис. 1, иллюстрирует компоновку метаданных научной статьи в соответствии с этим стандартом.

Согласно стандартам, разработанным EuDML [7–10], наборы данных различаются по степени необходимости их включения в XML-документ. Перечислим их.

Обязательные (Mandatory) – это элементы, которые должны присутствовать в каждом JATS-документе. Они необходимы, чтобы документ соответствовал минимальным требованиям стандарта. Примеры обязательных элементов включают `<article-id>` (уникальный идентификатор статьи), `<article-title>` (название на языке оригинала) и `<contrib-group>` (список авторов).

```
<article>
    <front>
        <article-meta>
            <title-group>
                <article-title>Название статьи</article-title>
            </title-group>
            <contrib-group>
                <contrib contrib-type="author">
                    <name>
                        <surname>Хамеджанов</surname>
                        <given-names>A.P.</given-names>
                    </name>
                    <xref ref-type="aff" rid="aff0"/>
                </contrib>
                <!-- Остальные авторы -->
            </contrib-group>
            <!-- Аффиляция -->
            <aff id="aff1">
                <institution>Название университета</institution>
                <addr-line>Адрес университета</addr-line>
                <country>Страна</country>
            </aff>
            <abstract>
                <p>Аннотация статьи...</p>
            </abstract>
            <kwd-group>
                <kwd>Ключевое слово 1</kwd>
                <kwd>Ключевое слово 2</kwd>
                <!-- Дополнительные ключевые слова -->
            </kwd-group>
        </article-meta>
    </front>
</article>
```

Рис. 1. Фрагмент XML-файла, созданного согласно схеме NISO JATS V1.0.

Фундаментальные (Fundamental) – эти элементы считаются основными для структуры и содержания научной статьи, но не всегда обязательны для каждой статьи. Они включают такие элементы, как <abstract> (аннотация) и <kwd-group> (ключевые слова), которые предоставляют основную информацию о статье и ее содержании.

Дополнительные (Supplemental) – это элементы, которые могут быть добавлены в JATS-документ для обогащения информации, но не являются необходимыми для соответствия стандарту. Примеры включают `<ref-list>` (список литературы), `<funding-group>` (данные о финансировании исследования), `<ext-link>` (различные ссылки с дополнительной информацией) и др., которые могут предоставлять дополнительные данные о статье или ее авторах.

В работе [11] отмечено, что схемы, предложенные EuDML, не позволяют в рамках единого набора метаданных описать статью, опубликованную на русском языке в научном журнале, и ее перевод в англоязычной версии этого журнала. Рекомендованы расширенная xml-схема, учитывающая указанную особенность, и соответствующие алгоритмы нормализации метаданных.

Данные разграничения блоков метаданных также актуальны для текущей разрабатываемой системы. Настоящая работа является продолжением исследований, представленных в [12].

В первой части содержится краткий анализ близких по тематике научных исследований. Во второй части описаны модель работы системы, спроектированная архитектура проекта и алгоритмы извлечения, дополнения и уточнения метаданных. В последней части указаны примеры метаданных научных документов, которые можно получить при обращении к различным семантическим сетям.

1. ИССЛЕДОВАНИЯ, БЛИЗКИЕ ПО ТЕМАТИКЕ

В работе [13] предложен алгоритм автоматического формирования метаданных выпусков научного журнала для экспорта в международные информационно-аналитические системы.

В статьях [14–17] предложен метод уточнения и дополнения аффилиации авторов с использованием семантической сети WikiData. Разработаны алгоритмы извлечения аффилиации из текста и дополнения полученных метаданных через SPARQL-запросы к базе знаний WikiData. Предложенные методы позволяют автоматически уточнять информацию об организациях – местах работы авторов статей, включая названия, адреса, страны и другие атрибуты.

Как правило, ядром экосистемы цифровых библиотек является фабрика метаданных (см., например, [18, 19]). В работе [19] приведено следующее определение этого термина: фабрика метаданных – это система взаимосвязанных программных инструментов, направленных на создание, обработку, хранение и управление метаданными объектов цифровых библиотек и позволяющих интегрировать создаваемые электронные коллекции в агрегирующие цифровые научные библиотеки.

В статьях [20, 21] представлена фабрика метаданных цифровой математической библиотеки Lobachevskii-DML [22, 23]. Эта фабрика включает набор сервисов для автоматизированного формирования, обработки и верификации метаданных; реализованы также программные инструменты нормализации метаданных в форматы агрегирующих библиотек [24]. Система, представленная в настоящем исследовании, разрабатывалась в соответствии со структурой и схемами указанной фабрики метаданных.

В работах [25, 26] предложены методы автоматической обработки научных документов, основанные на анализе структуры документов и применении методов семантического анализа. В частности, разработан метод извлечения из документа именованных сущностей с использованием предметных онтологий, что позволяет расширить набор ключевых слов документа. В [27, 28] представлена система сервисов автоматической обработки больших коллекций научных документов: извлечение метаданных из документов коллекции производится на основе анализа их структуры и форматов представления информации; созданные сервисы используют онтологии описания структуры документов [29].

В [30, 31] предложены решения основных задач, связанных с формированием цифровых математических коллекций из документов, изданных в доцифровой период, – такие коллекции обозначены авторами как ретроколлекции. Разработаны алгоритмы создания метаописания ретроколлекций, основанные на анализе структуры математических документов и применении программных инструментов выделения метаданных. Представлено описание ретроколлекций, сформированных с помощью разработанных алгоритмов и включенных в состав фабрики метаданных цифровой математической библиотеки Lobachevskii-DML. Указаны схемы формирования метаданных и методы нормализации извлеченной информации в соответствии со схемами и требованиями интегрирующих математических библиотек.

2. МОДЕЛЬ РАБОТЫ ПРОГРАММНОГО РЕШЕНИЯ

Процесс работы созданной программы можно разбить на отдельные этапы, где каждый следующий этап выполняется на основе результата предыдущего. Принцип работы системы в общем виде представлен на рис. 2 в виде UML-диаграммы деятельности.

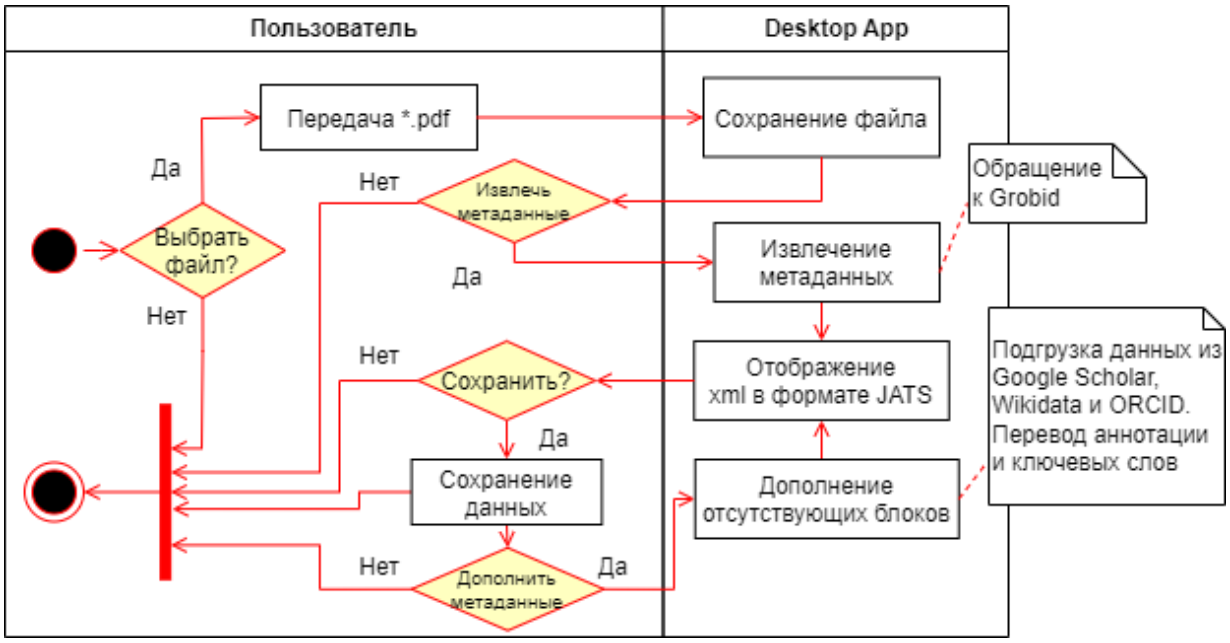


Рис. 2. Модель работы плагина в виде UML-диаграммы деятельности.

В общем виде работу системы можно описать следующим образом: пользователь подает на вход научный документ в виде pdf-файла, далее он может воспользоваться функцией извлечения метаданных, которая проанализирует документ и выделит блоки метаданных с помощью программного сервиса Grobid (GeneRation Of BIbliographic Data, https://grobid.readthedocs.io/en/latest/Introduction) и выведет данные в формате NISO JATS. На следующем шаге пользователю доступна функция дополнения метаданных, которая проанализирует ранее полученный xml-файл и выполнит поиск данных в Google Scholar, WikiData и ORCID, а также осуществит перевод аннотации и ключевых слов с помощью Yandex Translate. Полученные результаты оформляются в единый xml-файл и выводятся на экран. На каждом этапе работы пользователь может сохранить полученный промежуточный результат.

Разработанный алгоритм извлечения метаданных из научной работы включает функциональную последовательность операций (рис. 3). Сначала выбираются параметры извлечения и сохраняется файл в системе для дальнейшей передачи. Затем создается и отправляется запрос к сервису извлечения метаданных для обработки файла согласно заданным настройкам с помощью специального http-клиента Ktor (https://ktor.io). Далее данные, полученные в формате TEI (Text Encoding Initiative), преобразуются в соответствии с XML-схемой NISO JATS и специальным XSL-файлом (eXtensible Stylesheet Language) в XML-файл.

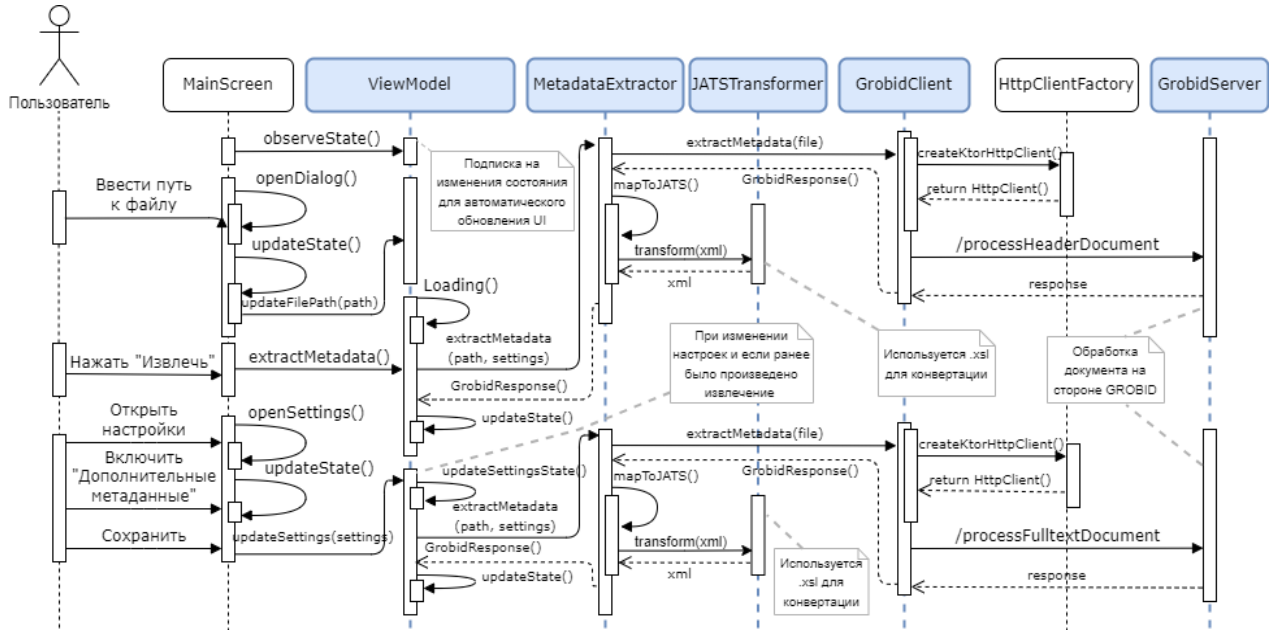


Рис. 3. Диаграмма последовательности извлечения метаданных из научного документа с настройкой необходимости формирования и включения в набор дополнительных метаданных.

Общая схема процесса дополнения и уточнения блоков метаданных состоит из нескольких шагов и представлена в виде псевдокода ниже (см. Алгоритм 1). На первом этапе выполняется поиск метаданных по названию статьи и списку авторов в поисковой системе Google Scholar. Далее проводится обогащение полученной информации посредством обращения к базе знаний WikiData. На следующем шаге осуществляется поиск по названию статьи в реестре уникальных идентификаторов авторов ORCID. В случае обнаружения кода ORCID автора при извлечении метаданных из документа либо на предыдущих шагах выполняется обращение к реестру ORCID для уточнения и дополнения информации

об авторе. Завершающим этапом являются анализ и сбор блоков метаданных в единый XML-файл согласно схеме NISO JATS.

Алгоритм 1: Дополнение и уточнение блоков метаданных из открытых источников

```
# Получение данных по извлеченному названию научной статьи
article_data = google_scholar.search_by_title(article_title)
# Формирование единого множества авторов
authors_set = parsed_authors.merge()
# Поиск данных по каждому автору согласно списку ФИО и id профилей авторов
author_profiles = []
for author in authors_set:
   if author.profile_id != null:
      author_details = google_scholar.get_author_profile(author.profile_id)
      author_profiles.add(author_details)
   else:
      author_profiles.add(google_scholar.search_author(author.full_name))
end for
# Получение данных для цитирования по уникальному идентификатору статьи
citation_data = google_scholar.get_citation_data(article_data.article_id)
# Объединение метаданных в единую модель
scholar_data = merge(article_data, author_profiles, citation_data)
scholar_data = filter_by_settings(scholar_data, user_settings)

# Обогащение полученной информации через WikiData
wikidata_data = []
for author in authors_set:
   wikidata_data.add(wikidata.query(author))
end for
wikidata_data = filter_by_settings(wikidata_data, user_settings)

# Поиск по названию статьи в реестре ORCID
orcid_search_results = orcid.search_by_title(article_title)
# Формирование единого множества найденных ORCID авторов
orcid_set = find_orcid(authors_set, orcid_search_results)
orcid_authors = []
for id in orcid_set:
```

```
    # Обращение к реестру ORCID для уточнения и дополнения данных об авторе
    orcid_authors.add(orcid.get_record(id))
end for
# Фильтрация согласно настройкам пользователя (включение/исключение дополнительных метаданных)
orcid_authors = filter_by_settings(orcid_authors, user_settings)

# Анализ и сбор блоков метаданных в единый XML-файл согласно схеме NISO JATS
result = merge(scholar_data, wikidata_data, orcid_search_results, orcid_authors)
jats_xml = format_result(result)
```

Для перевода аннотации и ключевых слов статьи с русского языка на английский (или наоборот) реализован модуль перевода с помощью сервиса Yandex Translate API. Разработанный алгоритм перевода представлен ниже в виде псевдокода (см. Алгоритм 2). Сначала выполняется запрос, содержащий название статьи, к Yandex Translate API для определения языка текста. Далее отправляются два запроса с текстом аннотации и списком ключевых слов. Полученные результаты перевода интегрируются в выходной XML-файл согласно стандарту NISO JATS V1.0.

Алгоритм 2. Перевод аннотации и ключевых слов статьи

```
# Разбор XML-дерева и получение названия статьи
load article_title = input.xml
# Разбор XML-дерева и получение текста аннотации
load abstract = input.xml
# Разбор XML-дерева и получение списка ключевых слов
load kwds = input.xml
# Определение языка по названию статьи
article_lang = yandex_api.detect(article_title)
# Перевод на английский, если текст написан на русском
if article_lang == 'ru':
    # Конфигурация и отправка запроса перевода на английский
    if abstract != "":
        abstract_translated = translate_en(abstract)
    for kwd in kwds:
```

```
    if kwd != "":
      kwds_translated = translate_en(kwds)
  end for
# Перевод на русский, если текст написан на английском
if article_lang == 'en':
  # Конфигурация и отправка запроса перевода на русский
  if abstract != "":
    abstract_translated = translate_ru(abstract)
  for kwd in kwds:
    if kwd != "":
      kwds_translated = translate_ru(kwds)
  end for
# Интеграция результатов с языковой меткой в xml файл

add_translation(article_lang, abstract_translated, kwds_translated)
```

Архитектура разработанной системы представлена в виде диаграммы компонентов, которая описывает связи внутри программного решения (рис. 4). Такое архитектурное решение убирает связанность между модулями, что, в свою очередь, позволяет в дальнейшем интегрировать дополнительные источники данных для увеличения полноты и точности формируемых метаданных.

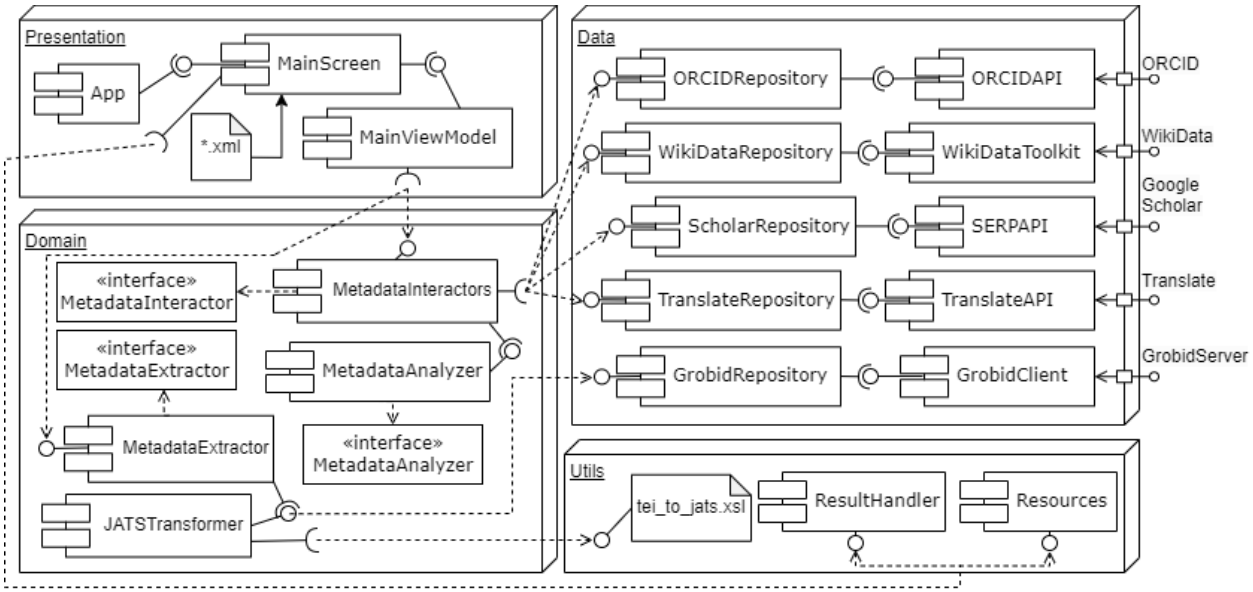


Рис. 4. Диаграмма компонентов разработанной системы автоматического формирования блока метаданных научных документов с использованием открытых баз данных.

3. РЕЗУЛЬТАТЫ

На рис. 5–7 представлен пример фрагмента ответа от сервиса Google Scholar SERP API в формате JSON (ссылки и аннотации были сокращены для читабельности примеров). Согласно этим данным (рис. 5) можно получить аффилиации, фотографию из профиля автора и список ключевых тематик работ автора, который в дальнейшем может помочь решить проблему отличия полных тезок. Кроме того можно расширить блок дополнительных метаданных ссылкой на публикацию, названием журнала и издательской компании, годом публикации, аннотацией и отдельной ссылкой на полный текст статьи (рис. 6 и 7).


```
"profiles": [
  {
    "name": "Evgeny Lipachev",
    "author_id": "HWLef7EAAAAJ",
    "affiliations": "Kazan Federal University",
    "email": " elipachev@gmail.com",
    "cited_by": 1211,
    "interests": [
      {
        "title": "Веб-технологии"
      },
      {
        "title": "краевые задачи дифракции"
      },
      {
        "title": "электронные библиотеки"
      },
      {
        "title": "MathML"
      }
    ],
"thumbnail":"https://scholar.googleusercontent.com/citations?view
_op=small_photo&user=HWLef7EAAAAJ&citpid=2"
```


Рис. 5. Фрагмент ответа от сервиса Google Scholar SERP API при поиске по ФИО автора "Evgeny Lipachev".

```
      "title": "OntoMath PRO Ontology: A Linked Data Hub for Mathematics",
      "result_id": "BcazuB-eH48J",
      "link": "https://link.springer.com/chapter/10.1007/978-3",
      "snippet": "In this paper, we present an ontology of...",
      "publication_info": {
        "summary": "OA Nevzorova, N Zhiltsov, A Kirillovich… - … Engineering and
the …, 2014 - Springer",
       "authors": [{
           "name": "OA Nevzorova",
           "link":"/citations?user=n2GFYqkAAAAJr&hl=en&oi=sra,
           "author_id": "n2GFYqkAAAAJ"
         },//остальные авторы ]
      },
      "resources": [//ссылки для скачивания]
```


Рис. 6. Фрагмент ответа от сервиса Google Scholar SERP API при поиске по названию статьи "OntoMath PRO Ontology: A Linked Data Hub for Mathematics".


```
          "citations": [
              {
                "title": "MLA",
                   "snippet": "Nevzorova, Olga A., et al. \"OntoMath PRO
ontology: a linked data hub for mathematics.\" Knowledge Engineering
and the Semantic Web: 5th International Conference, KESW 2014, Kazan,
Russia, September 29-October 1, 2014. Proceedings 5. Springer
International Publishing, 2014."
              },
```


Рис. 7. Пример фрагмента ответа от сервиса Google Scholar SERP API запросе информации для цитирования статьи по ранее найденному уникальному идентификатору "BcazuB-eH48J" (см. рис. 5).

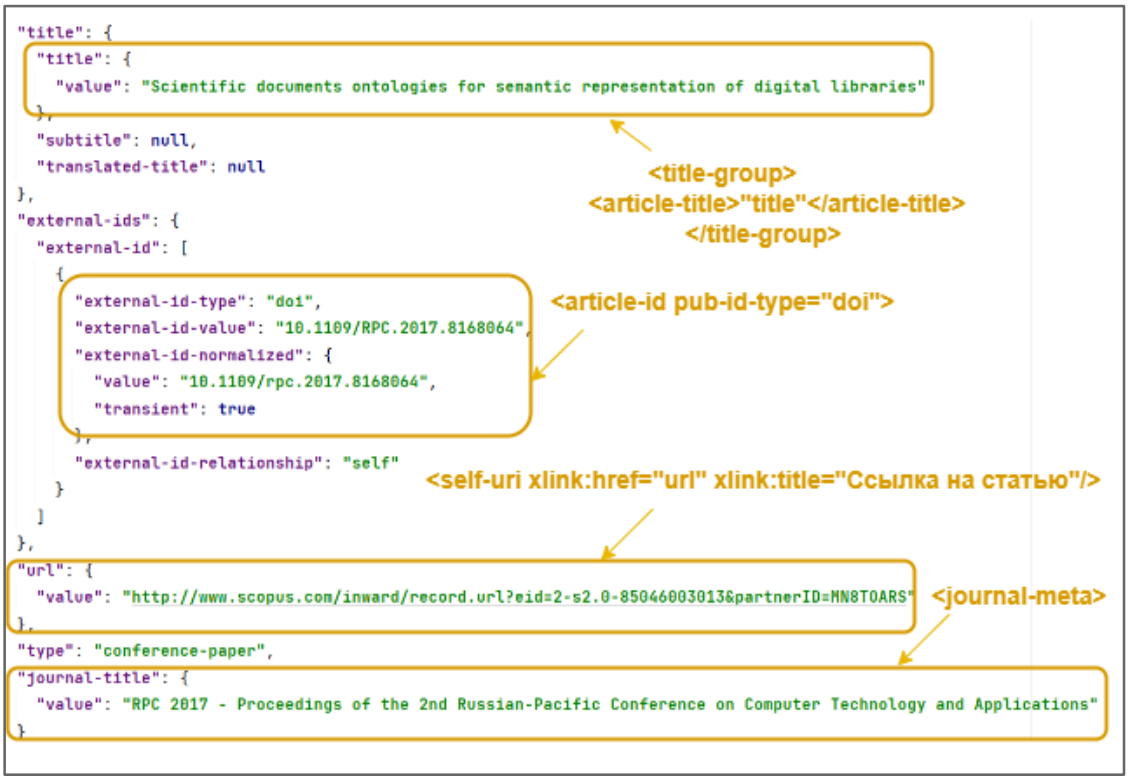


Рис. 8. Фрагмент ответа от сервиса ORCID API при запросе данных по коду автора «0000-0001-7789-2332».

Извлечение информации об авторе из реестра ORCID позволяет получить данные не только об авторе, но и о текущей статье, если в ней они присутствуют. Для этого выполняется поиск статьи среди списка публикаций автора, полученного на предыдущей шаге. Подобный подход позволяет установить блок дополнительных метаданных при их наличии, таких как DOI статьи, перевод заголовка статьи, ссылку на публикацию в журнале, дату публикации и название журнала (рис. 8).

ЗАКЛЮЧЕНИЕ

Благодаря взаимодействию с Google Scholar, ORCID и WikiData могут быть уточнены и дополнены аффилиации, ФИО, адрес электронной почты, код ORCID автора, а также дополнительные метаданные в виде ссылок на профили авторов в других научных сервисах, места работы и учебы авторов, списки научных статей, соответствующих ключевых слов, тематика научных работ. Многое зависит от степени открытости профиля авторов (https://info.orcid.org/privacy-policy),

а также от полноты информации, представленной на сайте ORCID. Работа с данным сервисом является полезным инструментом, так как в некоторых случаях подобная информация позволит не просто уточнить и дополнить метаданные, но и расширить круг поиска.

Основным результатом работы является разработка гибкой и расширяемой архитектуры системы, где каждый модуль инкапсулирует строго определенную задачу и может быть независимо заменен или модернизирован без нарушения целостности остальных компонентов. Дальнейшее направление развития заключается в совершенствовании отдельных модулей, отвечающих за методы обработки, извлечения метаданных и добавлении новых источников информации.

СПИСОК ЛИТЕРАТУРЫ

1. *Gartner R.* Metadata. Shaping Knowledge from Antiquity to the Semantic Web. Springer Cham, 2016. https://doi.org/10.1007/978-3-319-40893-4

2. *Kogalovsky M.R.* Metadata in Computer Systems // Programming and Computer Software. 2013. V. 39, No 4. P. 182–193. https://doi.org/10.1134/S0361768813040038

3. *Xie I., Matusiak K.K.* Discover Digital Libraries Theory and Practice. Elsevier Inc., 2016.

4. *Когаловский М.Р.* Метаданные, их свойства, функции, классификация и средства представления // CEUR Workshop Proceedings. 2012. V. 934. P. 3–14.

5. *Когаловский М.Р., Серебряков В.А.* Метаданные // Общенациональный интерактивный энциклопедический портал «Знания». 2022. № 9. https://doi.org/10.54972/00000048_2022_9_48

6. *Olver P.J.* The World Digital Mathematics Library: Report of a Panel Discussion // Proceedings of the International Congress of Mathematicians, August 13–21, 2014, Seoul, Korea. Kyung Moon SA, 1. 2014. P. 773–785.

7. EuDML metadata schema specification (v2.0–final). URL: https://initiative.eudml.org/eudml-metadata-schema-specification-v20-final (дата доступа 14.03.2026)

8. The EuDML metadata schema. Revision: 1.6 as of 15th December 2010.

/ Jost M., Bouche T., Goutorbe C., Jorda J.P. URL: http://www.mathdoc.fr/publis/d3.2-v1.6.pdf (дата доступа 14.03.2026)

9. *Sylwestrzak W., Borbinha J., Bouche T., Nowiński A., Sojka P.* EuDML – Towards the European Digital Mathematics Library // In: SojkaP. (Ed.) Towards a Digital Mathematics Library. Masaryk University, 2010. P. 11–26.
URL: https://eudml.org/doc/220786 (дата доступа 14.03.2026)

10. *Bouche T.* Reviving the free public scientific library in the digital age? The EuDML project // In: Kaiser K., Krantz S.G., Wegner B. (Eds.) Topics and Issues in Electronic Publishing JMM/AMS Special Session, FIZ Karlsruhe. 2013. P. 57–80.
URL: https://www.emis.de/proceedings/TIEP2013/05bouche.pdf (дата доступа 14.03.2026)

11. *Gafurova P.O., Elizarov A.M., Lipachev E.K., Khammatova D.M.* Metadata Normalization Methods in the Digital Mathematical Library // CEUR Workshop Proceedings. 2020. V. 2543. P. 136–148.

12. *Хамеджанов А.Р.* Система автоматического формирования блока метаданных научных документов с использованием открытых баз данных // Системы высокой доступности. 2026. Т. 22, № 1. С. 51−55.
https://doi.org/ 10.18127/j20729472-202601-10

13. *Герасимов А.Н., Елизаров А.М., Липачев Е.К.* Формирование метаданных для международных баз цитирования в системе управления электронными научными журналами // Электронные библиотеки. 2015. Т. 18, № 1–2. С. 6–31.

14. *Гафурова П.О., Липачев Е.К.* Метод уточнения аффилиации авторов научных документов на основе запросов к семантической сети // Научный сервис в сети Интернет: труды XXIV Всероссийской научной конференции. М.: ИПМ им. М.В. Келдыша, 2022. С. 115–127. https://doi.org/10.20948/abrau-2022-31

15. *Gafurova P., Elizarov A., Lipachev E.* Algorithms for Integration of Unstructured Mathematical Documents into the Common Digital Space of Scientific Knowledge // Proc. Int. Conf. «Common Digital Space of Scientific Knowledge: Problems & Solutions« (CDSSK–2020). Moscow, Russia, November 10–12, 2020. CEUR Workshop Proceedings. 2021. V. 2990. P. 39–49.
http://ceur-ws.org/Vol-2990/rpaper4.pdf

16. *Elizarov A., Gafurova P., Lipachev E.* Wikidata in Metadata Formation

Methods for Documents of Digital Mathematical Library // CEUR Workshop Proceedings. 2021. V. 3066. P. 23–33.

17. *Гафурова П.О., Елизаров А.М., Липачев Е.К.* Извлечение знаний из Wikidata для формирования метаданных документов электронных математических коллекций // Электронные библиотеки. 2021. Т. 24, № 6. С. 1023–1059. https://doi.org/10.26907/1562-5419-2021-24-6-1023-1059

18. *Bouche T., Labbe O.* The New Numdam Platform // In: Geuvers H., England M., Hasan O., Rabe F., Teschke O. (Eds.) Intelligent Computer Mathematics. CICM 2017. Lecture Notes in Computer Science. Vol. 10383. Springer, Cham, 2017. P. 70–82. https://doi.org/10.1007/978-3-319-62075-6_6

19. *Елизаров А.М., Липачев Е.К.* Цифровые платформы и цифровые научные библиотеки // International Journal of Open Information Technologies. 2020. Т. 8. № 11. С. 80–90.

20. *Elizarov A., Lipachev E.* Digital Library Metadata Factories // CEUR Workshop Proceedings. 2021. V. 2813. P. 13–21.

21. *Гафурова П.О., Елизаров А.М., Липачев Е.К.* Базовые сервисы фабрики метаданных цифровой математической библиотеки Lobachevskii-DML // Электронные библиотеки. 2020. Т. 23, № 3. С. 336–381. https://doi.org/10.26907/1562-5419-2020-23-3-336-381

22. *Елизаров А.М., Липачев Е.К.* Цифровая библиотека Lobachevskii-DML в научном пространстве математических знаний // Научно-техническая информация. Серия 1: Организация и методика информационной работы. 2023. № 1. С. 32–37. https://doi.org/10.36535/0548-0019-2023-01-3

23. *Elizarov A., Lipachev E.* BIG MATH Methods in Lobachevskii-DML Digital Library // CEUR Workshop Proceedings. 2019. V. 2523. P. 59–72.

24. *Gafurova P.O., Elizarov A.M., Lipachev E.K., Khammatova D.M.* Metadata Normalization Methods in the Digital Mathematical Library // CEUR Workshop Proceedings. 2020. V. 2543. P. 136–148

25. *Biryal'tsev E., Elizarov A., Zhil'tsov N., Lipachev E., Nevzorova O., Solov'ev V.* Methods for ANALYZING Semantic Data of Electronic Collections in Mathematics // Automatic Documentation and Mathematical Linguistics. 2014. V. 48, No. 2. P. 81–85. https://doi.org/10.3103/S000510551402006X

26. *Биряльцев Е.В., Елизаров А.М., Жильцов Н.Г., Липачев Е.К., Невзорова О.А., Соловьев В.Д.* Методы анализа семантических данных математических электронных коллекций // Научно-техническая информация. Серия 2: Информационные процессы и системы. 2014. № 4. С. 12–17.

27. *Elizarov A.M., Lipachev E.K., Khaydarov S.M.* Automated System of Services for Processing of Large Collections of Scientific Documents // CEUR Workshop Proceedings. 2016. V. 1752. P. 58–68.

28. *Elizarov A., Khaydarov S., Lipachev E.* Scientific Documents Ontologies for Semantic Representation of Digital Libraries // RPC 2017 – Proceedings of the 2nd Russian-Pacific Conference on Computer Technology and Applications. 2017. P. 1–5. https://doi.org/10.1109/RPC.2017.8168064

29. *Peroni S.* Semantic Web Technologies and Legal Scholarly Publishing. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-04777-5

30. *Андреичев М.Д., Гафурова П.О., Елизаров А.М., Липачев Е.К.* Пополнение метаданных документов математических цифровых ретро-коллекций методом семантических сетей // Научный сервис в сети Интернет: труды XXIII Всероссийской научной конференции (20-23 сентября 2021 г., онлайн). М.: ИПМ им. М.В.Келдыша, 2021. С. 22–33. https://doi.org/10.20948/abrau-2021-22

31. *Гафурова П.О., Елизаров А.М., Липачев Е.К.* Алгоритмы формирования метаданных математических ретро-коллекций на основе анализа структурных особенностей документов // Электронные библиотеки. 2021. Т. 24, № 2. С. 238–271. https://doi.org/10.26907/1562-5419-2021-24-2-238-270

THE SYSTEM FOR THE AUTOMATIC GENERATION, PROCESSING, AND MANAGEMENT OF DOCUMENT METADATA IN DIGITAL COLLECTIONS

A. R. Khamedzhanov [0009-0000-5624-2453]

Kazan (Volga region) Federal University, Kazan, Russia

hamedzhanovalmaz@gmail.com

Abstract

The publishing cycle is currently undergoing significant technological changes: automated publication management systems are being implemented, neural network technologies are being used for content processing, and tools for the intelligent analysis of scientific data are being actively developed. One of the key trends is the automation of the publishing cycle, aimed at accelerating manuscript processing, improving the quality of metadata, and ensuring the interoperability of information resources. In this context, metadata serves as a connecting element for machine processing and navigation within the scientific knowledge space, ensuring the structuring, interpretation, and integration of information into digital library systems. However, metadata for scientific publications often contain errors, inaccuracies, or are incomplete, and their manual creation and refinement are time-consuming and do not ensure high accuracy. The aim of this work is to design and develop a system for the automatic generation, processing, and management of metadata for scientific documents based on data obtained from scientific publication search services and open knowledge bases. The system can be used to automate the process of extracting, refining, and supplementing the metadata of scientific publications for the purpose of subsequently creating electronic collections of scientific documents.

Keywords*: digital mathematical library, semantic networks, automation of editorial processes, metadata generation, metadata extraction, metadata addition, NISO JATS, digital libraries.*

REFERENCES

1. *Gartner R.* Metadata. Shaping Knowledge from Antiquity to the Semantic Web. Springer Cham, 2016. https://doi.org/10.1007/978-3-319-40893-4

2. *Kogalovsky M.R.* Metadata in Computer Systems // Programming and Computer Software. 2013. V. 39, No. 4. P. 182–193.
https://doi.org/10.1134/S0361768813040038

3. *Xie I., Matusiak K. K.* Discover Digital Libraries Theory and Practice. Elsevier Inc., 2016.

4. *Kogalovsky M.R.* Metadata, their Properties, Functions and Classifications // CEUR Workshop Proceedings. 2012. V. 934. P. 3–14.

5. *Kogalovsky M.R., Serebryakov V.A.* Metadata // National Interactive Encyclopedia Portal "Knowledge". 2022. No. 9.
https://doi.org/10.54972/00000048_2022_9_48

6. *Olver P.J.* The World Digital Mathematics Library: Report of a Panel Discussion // Proceedings of the International Congress of Mathematicians, August 13–21, 2014, Seoul, Korea. Kyung Moon SA, 1. 2014. P. 773–785.

7. EuDML metadata schema specification (v2.0–final). URL: https://initiative.eudml.org/eudml-metadata-schema-specification-v20-final.

8. The EuDML metadata schema. Revision: 1.6 as of 15th December 2010. / Jost M., Bouche T., Goutorbe C., Jorda J.P.
URL: http://www.mathdoc.fr/ publis/d3.2-v1.6.pdf (last access 04.04.2026)

9. *Sylwestrzak W., Borbinha J., Bouche T., Nowiński A., Sojka P.* EuDML – Towards the European Digital Mathematics Library // In: SojkaP. (Ed.) Towards a Digital Mathematics Library. Masaryk University, 2010. P. 11–26.
URL: https://eudml.org/doc/220786 (last access 04.04.2026)

10. *Bouche T.* Reviving the free public scientific library in the digital age? The EuDML project // In: Kaiser K., Krantz S.G., Wegner B. (Eds.) Topics and Issues in Electronic Publishing JMM/AMS Special Session, FIZ Karlsruhe. 2013. P. 57–80.
URL: https://www.emis.de/proceedings/TIEP2013/05bouche.pdf (last access 04.04.2026)

11. *Gafurova P.O., Elizarov A.M., Lipachev E.K., Khammatova D.M.* Metadata Normalization Methods in the Digital Mathematical Library // CEUR Workshop Proceedings. 2020. V. 2543. P. 136–148.

12. *Khamedzhanov A.R.* The system of automatic generation of a block of metadata of scientific documents using open databases // Highly Available Systems. 2026. V. 22, No. 1. P. 51−55. https://doi.org/10.18127/j20729472-202601-10

13. *Gerasimov A.N., Elizarov A.M., Lipachev E.K*. Formation of metadata for international citation databases in the management system of electronic scientific journals // Russian Digital Libraries Journal. 2015. V. 18, No. 1–2. P. 6–31.

14. *Gafurova P.O., Lipachov E.K.* Method for Clarifying the Affiliation of Authors of Scientific Documents Based on Requests to the Semantic Web. XXIV All-Russian Scientific Conference 'Scientific Service on the Internet'. 2022. P. 115–127. https://doi.org/10.20948/abrau-2022-31

15. *Gafurova P., Elizarov A., Lipachev E.* Algorithms for Integration of Unstructured Mathematical Documents into the Common Digital Space of Scientific Knowledge // CEUR Workshop Proceedings. 2021. V. 2990. P. 39–49.
URL: http://ceur-ws.org/Vol-2990/rpaper4.pdf (last access 04.04.2026)

16. *Elizarov A., Gafurova P., Lipachev E.,* Wikidata in Metadata Formation Methods for Documents of Digital Mathematical Library // CEUR Workshop Proceedings. 2021. V. 3066. P. 23–33.

17. *Gafurova P.O., Elizarov A.M., Lipachev E.K.* Extraction of Wikidata Knowledge for the Metadata Formation for Documents of Electronic Mathematical Collections // Russian Digital Libraries Journal. 2021. V. 24, No. 6. P. 1023–1059.
https://doi.org/10.26907/1562-5419-2021-24-6-1023-1059

18. *Bouche T., Labbe O.* The New Numdam Platform // In: Geuvers H., England M., Hasan O., Rabe F., Teschke O. (Eds.) Intelligent Computer Mathematics. CICM 2017. Lecture Notes in Computer Science. Vol. 10383. Springer, Cham, 2017. P. 70–82. https://doi.org/10.1007/978-3-319-62075-6_6

19. *Elizarov A., Lipachev E.* Digital Platforms and Digital Scientific Libraries // International Journal of Open Information Technologies. 2020. V. 8, No. 11. P. 80–90.

20. *Elizarov A., Lipachev E.* Digital Library Metadata Factories // CEUR Workshop Proceedings. 2021. V. 2813. P. 13-21.

21. *Gafurova P.O., Elizarov A.M., Lipachev E.K.* Basic Services of Factory Metadata Digital Mathematical Library Lobachevskii-DML // Russian Digital Libraries Journal. 2020. V. 23, No. 3. P.336–381.
https://doi.org/10.26907/1562-5419-2020-23-3-336-381

22. *Elizarov A.M., Lipachev E.K.* Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge // Automatic Documentation and Mathematical

Linguistics Series 1: Organization and Methods of Information Work. 2023. No. 1. P. 32–37. https://doi.org/10.36535/0548-0019-2023-01-3

23. *Elizarov A., Lipachev E.* BIG MATH Methods in Lobachevskii-DML Digital Library // CEUR Workshop Proceedings. 2019. V. 2523. P. 59–72.

24. *Gafurova P.O., Elizarov A.M., Lipachev E.K., Khammatova D.M.* Metadata Normalization Methods in the Digital Mathematical Library // CEUR Workshop Proceedings. 2020. V. 2543. P. 136–148

25. *Biryal'tsev E., Elizarov A., Zhil'tsov N., Lipachev E., Nevzorova O., Solov'ev V.* Methods for Analyzing Semantic Data of Electronic Collections in Mathematics // Automatic Documentation and Mathematical Linguistics. 2014. V. 48, No. 2. P. 81–85. https://doi.org/10.3103/S000510551402006X

26. *Biryal'tsev E., Elizarov A., Zhil'tsov N., Lipachev E., Nevzorova O., Solov'ev V.* Methods for analyzing semantic data of mathematical electronic collections // Scientific and Technical Information. Series 2: Information Processes and Systems. 2014. No 4. P. 12–17.

27. *Elizarov A.M., Lipachev E.K., Khaydarov S.M.* Automated System of Services for Processing of Large Collections of Scientific Documents // CEUR Workshop Proceedings. 2016. V. 1752. P. 58–68.

28. *Elizarov A., Khaydarov S., Lipachev E.* Scientific Documents Ontologies for Semantic Representation of Digital Libraries // RPC 2017 – Proceedings of the 2nd Russian-Pacific Conference on Computer Technology and Applications. 2017. P. 1–5. https://doi.org/10.1109/RPC.2017.8168064

29. *Peroni S.* Semantic Web Technologies and Legal Scholarly Publishing. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-04777-5

30. *Andreichev M.D., Gafurova P.O., Elizarov A.M., Lipachev E.K.* Replenishment of Documents of Mathematical Digital Retro-collections by Searching in Semantic Web. XXIII All-Russian Scientific Conference 'Scientific Service on the Internet'. 2021. P. 22–33. https://doi.org/10.20948/abrau-2021-22

31. *Gafurova P.O., Elizarov A.M., Lipachev E.K.* Algorithms for Formation of Metadata Mathematical Retro Collections Based on Analysis of Structural Features of Documents // Russian Digital Libraries Journal. 2021. V. 24, No 2. P. 238–271. https://doi.org/10.26907/1562-5419-2021-24-2-238-270

СВЕДЕНИЯ ОБ АВТОРЕ

ХАМЕДЖАНОВ Алмаз Рустамович – аспирант Института информационных технологий и интеллектуальных систем Казанского федерального университета

Almaz Rustamovich KHAMEDZHANOV– postgraduate student at the Institute Institute of Information Technology and Intelligent Systems, Kazan Federal University

email: hamedzhanovalmaz@gmail.com

ORCID: 0009-0000-5624-2453

Материал поступил в редакцию 23 марта 2026 года

УДК 004

О ПРИМЕНИМОСТИ НЕЙРОСЕТЕЙ В ИЗДАТЕЛЬСКОМ ДЕЛЕ

С. И. Ширинбегзода[1] [0009-0003-7317-4722], Д. А. Шишкин[2] [0009-0008-9742-4489], Б. С. Усманов[3] [0009-0008-5078-7266], Н. М. Боргест[4] [0000-0003-2934-6198]

[1–4]*Самарский национальный исследовательский университет имени академика С. П. Королева, г. Самара, Россия*

[1]shirinbegzodasi@yandex.ru, [2]Cr1stra61@yandex.ru, [3]usmanov.studios@yandex.ru, [4]borgest@yandex.ru

Аннотация

В работе дана оценка границ применимости больших языковых моделей в редакционных задачах издательского процесса и установлен оптимальный формат взаимодействия между человеком и алгоритмическими системами.

Методологической основой исследования является сравнительный эксперимент, в рамках которого несколько популярных нейросетевых моделей (Alice AI, GigaChat, DeepSeek, Gemini и ChatGPT) выполнен статистический анализ контрольного текста на русском языке. Определялись количественные характеристики текста: числа слов, символов с пробелами и без пробелов, а также количества абзацев. Полученные результаты сопоставлялись с эталонными значениями, установленными с помощью текстового редактора MS Word, использующего детерминированный алгоритм подсчета символов.

Результаты эксперимента показали, что нейросетевые модели демонстрируют различную степень точности при выполнении задач количественного анализа текста. Основной причиной подобных ошибок являются архитектура больших языковых моделей и использование алгоритмов токенизации, которые разрывают прямую связь между символами и внутренним представлением текста в модели.

На основе полученных результатов предложена концепция гибридной архитектуры издательских информационных систем, в которой генеративные языковые модели используются для выполнения творческих и аналитических задач, а операции, требующие строгой формальной точности, передаются специализи-

рованным детерминированным микросервисам. Предложенный подход позволяет повысить надежность и предсказуемость работы интеллектуальных издательских систем.

***Ключевые слова**: искусственный интеллект, издательское дело, большие языковые модели, нейросети, автоматизация, токенизация, редакционный процесс.*

ВВЕДЕНИЕ

Искусственный интеллект (ИИ) активно трансформирует издательскую индустрию, автоматизируя процессы обработки текста и графики. Рост интереса к генеративным языковым моделям связан с быстрым развитием методов обработки естественного языка (Natural language processing, NLP) и появлением так называемых больших языковых моделей (Large Language Models, LLM), обученных на огромных корпусах текстовых данных. Такие системы способны генерировать связный текст, выполнять перевод, реферирование и ряд других задач обработки информации. Однако их применение в профессиональных производственных процессах, включая издательское дело, требует детального анализа надежности и точности их работы. Актуальность настоящего исследования обусловлена необходимостью понимать не только возможности, но и технические ограничения этих инструментов, чтобы избегать ошибок в производстве.

АНАЛИЗ СУЩЕСТВУЮЩИХ ИССЛЕДОВАНИЙ

Внедрение нейросетевых технологий в издательский цикл происходит неравномерно, затрагивая автоматизацию производства, прогнозную аналитику и изменение характера человеческого труда [1]. Наиболее ощутимо влияние ИИ в работе с текстом, где инструменты обработки естественного языка ускоряют корректуру, предлагают стилистические правки и создают черновые переводы [2]. Все это ведет не к замещению специалистов, а к смене парадигмы их деятельности: акцент смещается с рутинного исполнения на экспертную верификацию. Для редакторов и переводчиков приоритетом становится контроль контекста и культурных нюансов, тогда как для авторов нейросети выступают в роли генераторов идей, оставляя функцию смыслообразования за человеком. Однако

эффективность цифровых помощников ограничена техническими особенностями больших языковых моделей. Демонстрируя успехи в «креативных» задачах, алгоритмы часто оказываются несостоятельными в строгих формальных операциях [3].

Особо подчеркнем, что внедрение LLM в издательские дело требует строгого контроля, так как алгоритмы склонны к галлюцинациям и различным типам искажений [4]. При проектировании таких программно-аппаратных комплексов на первый план выходят требования к предсказуемости и высокой доступности ИИ как сервиса [5].

МЕТОДИКА ИССЛЕДОВАНИЯ

Одной из фундаментальных проблем современных генеративных моделей является их неспособность к точному подсчету количественных характеристик текста. Данный феномен, получивший в профессиональном сообществе условное название «проблема клубники» (от английского слова strawberry, в котором модели часто ошибочно насчитывают две буквы «r» вместо трех), имеет глубокие технические корни [6]. Причина систематических ошибок LLM кроется в их архитектуре: модели работают не с отдельными буквами, а с токенами – фрагментами слов, которые могут быть разной длины. Например, слово «деревья» в зависимости от словаря токенизатора может быть представлено как один токен или как два («*дерев*» + «*ья*»). При этом модель не хранит точную длину каждого токена в символах; ее ответ формируется на основе вероятностного предсказания, а не прямого вычисления [7].

Фундаментальное ограничение связано с алгоритмами токенизации, которые разрывают прямую связь между словом и составляющими его символами. Из-за этого понимание на уровне отдельных знаков формируется у моделей крайне медленно и нелинейно [8]. Некорректная токенизация является одной из главных «точек отказа», снижающих общую надежность LLM при обработке лингвистических запросов [9].

Для эмпирической проверки масштаба указанных погрешностей был проведен сравнительный эксперимент. В качестве эталонного инструмента был выбран текстовый редактор MS Word, который выполняет прямой посимвольный подсчет в кодировке Unicode. Это детерминированный алгоритм, не зависящий

от контекста и дающий абсолютно точный результат для заданной строки символов, что позволяет считать его «золотым стандартом» для измерения длины текста. Объектами исследования стали популярные нейросетевые модели: Alice AI, GigaChat, DeepSeek, Gemini и ChatGPT. Была поставлена задача провести полный статистический анализ контрольного текста на русском языке, а именно: подсчитать количество слов, знаков (с пробелами и без) и абзацев. Полученные результаты сопоставлялись с данными MS Word.

Все модели использовались через публичные интерфейсы без дополнительной настройки параметров генерации. Для каждой модели был выполнен отдельный анализ контрольного текста с использованием одинакового запроса, что обеспечивало сопоставимость результатов.

В качестве метрики точности бралась относительная погрешность вычислений. Она определялась как отношение абсолютного отклонения результата модели от эталонного значения к эталону: комбинированная функция потерь имеет следующий вид:

$$\delta = \frac{|X_{\text{model}} - X_{\text{ref}}|}{X_{\text{ref}}} \times 100,$$

где X_{model} – результат, полученный нейросетевой моделью; X_{ref} – эталонное значение, полученное в MS Word.

Использование этой метрики позволило количественно оценить степень отклонения результатов различных моделей от детерминированного алгоритма подсчета символов.

Для анализа был отобран фрагмент художественного текста на русском языке под условным названием «Осенний день», обладающий следующей структурой:

«*Осенний день*

В октябре природа готовится к зимнему сну. Деревья сбрасывают последние листья, которые кружатся в воздухе, словно золотые бабочки. Ветер гонит их по пустынным улицам, создавая причудливые узоры на асфальте.

Небо серое, низкое, но дождя нет. Солнце изредка пробивается сквозь тучи, освещая мокрые тротуары. В парках тишина, лишь изредка нарушаемая шорохом опавшей листвы.

Прохожие спешат по своим делам, кутаясь в шарфы. Дети собирают красивые листья для гербария. Воздух свежий, прозрачный, наполненный особым осенним ароматом.

Этот день – как прощальный подарок уходящей осени.»

В нейросети отправлялся следующий промпт: «Используй Python для подсчета: общее количество слов; количество знаков без пробелов; количество знаков с пробелами; количество абзацев. Текст: [далее следовал текст]». Эталонные количественные характеристики текста приведены на рис. 1.

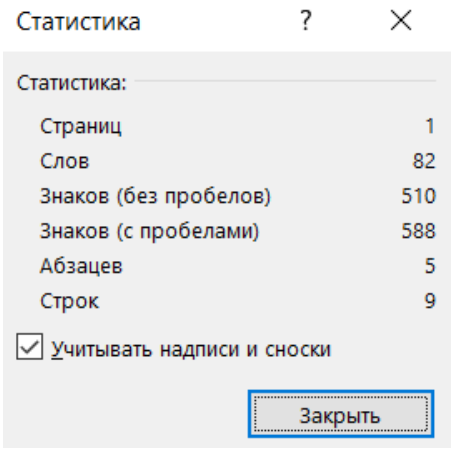


Рис. 1. Эталонные количественные характеристики текста.

ЭКСПЕРИМЕНТ И РЕЗУЛЬТАТЫ

Перед нейросетями была поставлена задача извлечь из контрольного текста ключевые метрические показатели. Сравнительный анализ (см. табл. 1) полученных данных позволил выявить существенные различия в точности работы алгоритмов.

Табл. 1. Результаты подсчета количественных характеристик текста различными нейросетевыми моделями (промт 1, данные актуальны на 26 января 2026 г.).

Модель/ инструмент	Слова, шт.	Знаков (без пробелов), шт.	Знаков (с пробелами), шт.	Абзацы, шт.	δ, % с пробелами	Ссылки на диалог
MS Word (эталон)	82	510	588	5	–	–
Alice AI	97	1072	1257	5	113.78	–
GigaChat	81	433	499	4	15.14	https://giga.chat/link/gcsbTRNPmY
Deepseek (Глубокое мышление)	97	511	608	4	3.40	https://chat.deepseek.com/share/muepx5r93pya7fvwe6
Gemini (Думающая)	83	502	588	5	0	https://gemini.google.com/share/a4dbfcf1e721
Gemini (Профессиональная)	84	516	603	5	2.55	https://gemini.google.com/share/ec4a97dbcb4f
ChatGPT	83	510	592	5	0.68	https://chatgpt.com/share/69775e41–b47c–8007–b548–09f519e5bd61

Как видно из табл. 1, большинство моделей демонстрирует значительные отклонения от эталонных значений. Наибольшая погрешность наблюдается у модели Alice AI, где число символов оказалось завышено более чем в два раза.

Лидером по точности стала модель ChatGPT, продемонстрировавшая минимальное отклонение от эталона (+0.68% знаков с пробелами) и верный подсчет абзацев. Сопоставимые результаты показала Gemini (Думающая). Осталь-

ные модели продемонстрировали значительные погрешности, что делает их непригодными для точного учета издательских объемов.

Полученные результаты подтверждают фундаментальную особенность LLM – отсутствие строгого механизма детерминированных вычислений. Даже при использовании инструкций, предполагающих выполнение программного кода, модель фактически генерирует вероятностный текстовый ответ, имитируя процесс вычисления. Это объясняет наблюдаемую нестабильность результатов.

Дополнительным фактором является различие в алгоритмах токенизации, используемых разными LLM. Поскольку слова могут разбиваться на различные токены, модель оперирует не символами, а вероятностными последовательностями токенов. Это приводит к накоплению ошибок при выполнении задач, требующих точного посимвольного анализа.

Кроме того, следует учитывать, что некоторые модели оптимизированы преимущественно для генерации связного текста, а не для выполнения формальных аналитических операций. В результате они демонстрируют высокие результаты в задачах генерации, но уступают детерминированным алгоритмам при точных вычислениях.

Для проверки устойчивости полученных результатов был проведен дополнительный эксперимент с альтернативной формулировкой запроса (см. табл. 2). В частности, был использован следующий запрос: «Проанализируй текст и выведи статистику: количество слов, знаков без пробелов, знаков с пробелами, абзацев. Не пиши код, используй свои внутренние возможности подсчета».

Табл. 2. Результаты подсчета количественных характеристик текста различными нейросетевыми моделями (промт 2, данные актуальны на 26 января 2026 г.)

Модель/ Инструмент	Слова, шт.	Знаков (без пробелов), шт.	Знаков (с пробелами), шт.	Абзацы, шт.	δ, % с пробелами)	Ссылки на диалог
MS Word (эталон)	82	510	588	5	–	–
Alice AI	107	602	725	5	23.30	–
GigaChat	86	423	481	5	18.20	https://giga.chat/link/gcs-COdLwUy
Deepseek (Глубокое мышление)	92	424	514	3	12.59	https://chat.deepseek.com/share/5v44zi6mr56qx214zz
Gemini (Думающая)	82	519	597	5	1.53	https://gemini.google.com/share/9964a7488c00
Gemini (Профессиональная)	82	517	593	5	0.85	https://gemini.google.com/share/d0e2febe1074
ChatGPT	82	511	588	5	0.00	https://chatgpt.com/share/69775bf3–6dd4–8007–87aa–e251d875375f

АРХИТЕКТУРНЫЕ ПРИНЦИПЫ

Попытки создать полностью автономные системы на базе LLM в издательской сфере часто сталкиваются с проблемами детерминированности и стабильности работы [10]. Для обеспечения высокой доступности издательских сервисов при пиковых нагрузках необходимо применять паттерны распределенных систем. В частности, интеграция LLM должна осуществляться через специализированные AI-шлюзы (AI Gateways), обеспечивающие механизмы автоматического выключения и альтернативной стратегии на случай отказа или некорректного ответа модели [11].

Как показал эксперимент, вероятностная природа LLM делает их неприменимыми для точных количественных операций. Поэтому в архитектуре высокодоступной системы управления контентом (Content Management System, CMS) должен применяться паттерн гибридной маршрутизации: задачи строго количественного анализа (уровень С) должны направляться к классическим детерминированным микросервисам, а творческие задачи – к LLM [12].

На основе полученных результатов может быть предложена концептуальная архитектура гибридной издательской системы. В такой системе интеграция LLM осуществляется через промежуточный слой – AI Gateway, который выполняет маршрутизацию запросов между различными типами сервисов.

Запросы, связанные с генерацией текста, стилистической правкой или реферированием, направляются к языковой модели. В то же время задачи, требующие строгой формальной точности (например, подсчет объема текста, проверка структуры документа или анализ ссылок) обрабатываются специализированными детерминированными микросервисами.

Подобная архитектура позволяет объединить преимущества генеративных моделей и классических алгоритмов, обеспечивая одновременно высокую производительность системы и необходимый уровень надежности.

ЗАКЛЮЧЕНИЕ

Проведенное исследование не только подтверждает необходимость участия человека в издательском процессе с использованием ИИ, но и позволяет уточнить границы применимости LLM. На основе анализа ошибок предлагается классифицировать редакционные задачи по трем уровням автономии:

- уровень A (полная автоматизация): задачи, не требующие высокой точности;
- уровень B (автоматизация с верификацией): задачи, где допустима небольшая погрешность, но результат должен быть проверен человеком (стилистическая правка, реферирование);
- уровень C (только инструментальный контроль): задачи, где ошибки LLM критичны (подсчет объема, проверка ссылок).

Перспективным направлением дальнейших исследований является разработка специализированных гибридных архитектур, сочетающих возможности LLM и традиционных алгоритмов обработки текста. В частности, актуальными являются задачи повышения надежности LLM-сервисов, создания систем автоматической верификации результатов и разработки методов интеграции языковых моделей в распределенные издательские платформы.

Развитие подобных систем позволит более эффективно использовать потенциал ИИ в издательской индустрии, одновременно минимизируя риски, связанные с вероятностной природой генеративных моделей.

Таким образом, будущее издательского дела – не в замене человека машиной, а в построении гибридных систем, где ИИ выступает ассистентом, берущим на себя рутину, а человек остается экспертом качества, особенно в задачах, требующих формальной точности. Такой подход позволяет не только избежать ошибок, но и повысить общую эффективность производства.

СПИСОК ЛИТЕРАТУРЫ

1. *Ryzkho O., Krainikova T., Vodolazka S., Sokolova K.* Generative AI changes the book publishing industry: reengineering of business processes // Communication and Society. 2024. V. 37 (3), P. 255–271.
https://doi.org/10.15581/003.37.3.255-271

2. Spubl. Использование искусственного интеллекта при написании научной статьи // Spubl. 2024. URL: https://spubl.com.ua/ru/blog/using-artificial-intelligence-when-writing-a-scientific-article (дата доступа 19.04.2026)

3. *Мурзин А.А.* Нейросети в книгоиздании: кейс «Интерактивная энциклопедия для школьников» // Известия Уральского федерального университета. Серия 1: Проблемы образования, науки и культуры. 2024. Т. 30, № 4. С. 165–173.

https://doi.org/10.15826/izv1.2024.30.4.072

4. *Ahn S.* The transformative impact of large language models on medical writing and publishing: current applications, challenges and future directions // Korean J Physiol Pharmacol. 2024. V. 28 (5). P. 393–401. https://doi.org/10.4196/kjpp.2024.28.5.393

5. Ensuring AI Reliability: Correctness, Consistency, and Availability. URL: https://dev.to/kapusto/ensuring-ai-reliability-correctness-consistency-and-availability-349p (дата доступа 19.04.2026)

6. *Xu N., Ma X.* LLM The Genius Paradox: A Linguistic and Math Expert's Struggle with Simple Word-based Counting Problems // Proc. of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL). 2025. https://doi.org/10.18653/v1/2025.naacl-long.172

7. Как этично использовать искусственный интеллект в написании научных статей. URL: https://a-articles.kz/iivnauchnyhstateyah/ (дата доступа 19.04.2026)

8. *Cosma A., Ruseti S., Radoi E., Dascalu M.* The Strawberry Problem: Emergence of Character-level Understanding in Tokenized Language Models // arXiv:2505.14172. 2025. https://doi.org/10.48550/arXiv.2505.14172

9. *Wang D. et al.* Tokenization Matters! Degrading Large Language Models through Challenging Their Tokenization // arXiv:2405.17067. 2024. https://doi.org/10.48550/arXiv.2405.17067

10. *Kamal I.* Why agentic LLM systems fail: Control, cost, and reliability // The New Stack. 2026. URL: https://thenewstack.io/why-agentic-llm-systems-fail-control-cost-and-reliability/ (дата доступа 19.04.2026)

11. *Gui C.* Best Practices for High Availability of LLM Based on AI Gateway // Alibaba Cloud Community. 2025. URL: https://www.alibabacloud.com/blog/best-practices-for-high-availability-of-llm-based-on-ai-gateway_602522 (дата доступа 19.04.2026)

12. *Topuz A.S.* LLM Integration in Distributed Systems: Engineering for Reliability at Scale // Medium (Software Engineering). 16.02.2026. https://dev.to/topuzas/llm-integration-in-distributed-systems-engineering-for-reliability-at-scale-l79 (дата доступа 19.04.2026).

ON THE APPLICABILITY OF NEURAL NETWORKS IN THE PUBLISHING INDUSTRY

S. I. Shirinbegzoda[1] [0009-0003-7317-4722], D. A. Shishkin[2] [0009-0008-9742-4489], B. S. Usmanov[3] [0009-0008-5078-7266], N. M. Borgest[4] [0000-0003-2934-6198]

[1–4]*Samara National Research University, Samara, Russia*

[1]shirinbegzodasi@yandex.ru, [2]Cr1stra61@yandex.ru, [3]usmanov.studios@yandex.ru, [4]borgest@yandex.ru

Abstract

The paper assesses the limits of applicability of large language models in editorial tasks within the publishing process and identifies the optimal format of interaction between humans and algorithmic systems.

The methodological basis of the study is a comparative experiment in which several popular neural network models — Alice AI, GigaChat, DeepSeek, Gemini, and ChatGPT — performed a statistical analysis of a control text in Russian. The quantitative characteristics of the text were determined: the number of words, characters with and without spaces, and the number of paragraphs. The obtained results were compared with reference values established using the MS Word text editor, which applies a deterministic character-counting algorithm.

The results of the experiment showed that neural network models demonstrate varying degrees of accuracy when performing tasks of quantitative text analysis. The main reason for such errors lies in the architecture of large language models and the use of tokenization algorithms, which break the direct connection between characters and the model's internal representation of the text.

Based on the results obtained, the paper proposes the concept of a hybrid architecture for publishing information systems, in which generative language models are used to perform creative and analytical tasks, while operations requiring strict formal accuracy are assigned to specialized deterministic microservices. The proposed approach makes it possible to improve the reliability and predictability of intelligent publishing systems.

Keywords*: artificial intelligence, publishing industry, large language models, neural networks, automation, tokenization, editorial workflow.*

REFERENCES

1. *Ryzkho O., Krainikova T., Vodolazka S., Sokolova K.* Generative AI changes the book publishing industry: reengineering of business processes // Communication and Society. 2024. V. 37 (3), P. 255–271.
https://doi.org/10.15581/003.37.3.255-271

2. Spubl. Использование искусственного интеллекта при написании научной статьи // Spubl. 2024. URL: https://spubl.com.ua/ru/blog/using-artificial-intelligence-when-writing-a-scientific-article (accessed 19.04.2026)

3. Murzin A.A. Neural networks in book publishing: the case of "Interactive encyclopedia for schoolchildren" // Izvestia Ural Federal University Journal. Series 1. Issues in Education, Science and Culture. 2024. V. 30, No. 4. P. 165–173.

4. *Ahn S.* The transformative impact of large language models on medical writing and publishing: current applications, challenges and future directions // Korean J Physiol Pharmacol. 2024. V. 28 (5). P. 393–401.
https://doi.org/10.4196/kjpp.2024.28.5.393

5. Ensuring AI Reliability: Correctness, Consistency, and Availability. URL: https://dev.to/kapusto/ensuring-ai-reliability-correctness-consistency-and-availability-349p (accessed 19.04.2026)

6. *Xu N., Ma X.* LLM The Genius Paradox: A Linguistic and Math Expert's Struggle with Simple Word-based Counting Problems // Proc. of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL). 2025. https://doi.org/10.18653/v1/2025.naacl-long.172

7. A–Articles. Как этично использовать искусственный интеллект в написании научных статей // A–Articles. 2025.
URL: https://a-articles.kz/iivnauchnyhstateyah/ (accessed 19.04.2026)

8. *Cosma A., Ruseti S., Radoi E., Dascalu M.* The Strawberry Problem: Emergence of Character-level Understanding in Tokenized Language Models // arXiv:2505.14172. 2025. https://doi.org/10.48550/arXiv.2505.14172

9. *Wang D. et al.* Tokenization Matters! Degrading Large Language Models through Challenging Their Tokenization // arXiv:2405.17067. 2024.
https://doi.org/10.48550/arXiv.2405.17067

10. *Kamal I.* Why agentic LLM systems fail: Control, cost, and reliability // The New Stack. 2026. URL: https://thenewstack.io/why-agentic-llm-systems-fail-control-

cost-and-reliability/ (accessed 19.04.2026)

11. *Gui C.* Best Practices for High Availability of LLM Based on AI Gateway // Alibaba Cloud Community. 2025. URL: https://www.alibabacloud.com/blog/best-practices-for-high-availability-of-llm-based-on-ai-gateway_602522 (accessed 19.04.2026).

12. *Topuz A.S.* LLM Integration in Distributed Systems: Engineering for Reliability at Scale // Medium (Software Engineering). 16.02.2026. https://dev.to/topuzas/llm-integration-in-distributed-systems-engineering-for-reliability-at-scale-l79 (accessed 19.04.2026)

СВЕДЕНИЯ ОБ АВТОРАХ

ШИРИНБЕГЗОДА Сухайлии Илхом – Студент 1 курса магистратуры направления «Авиастроение» и инженер Самарского национального исследовательского университета имени академика С. П. Королева. Специализируется в области авиастроения. Его научные интересы относятся к таким областям, как авиастроение, техника, технология, журналистика, AI-технологии, техническое зрение и образовательные платформы.

Suhaylii Ilhom SHIRINBEGZODA – first-year master's student in Aircraft Engineering and Engineer at Samara National Research University. He specializes in the field of aircraft engineering. His research interests include aircraft engineering, engineering science, technology, journalism, AI technologies, computer vision, and educational platforms.

email: shirinbegzodasi@yandex.ru

ORCID: 0009-0003-7317-4722

ШИШКИН Даниил Андреевич – Студент 1 курса магистратуры направления «Авиастроение» и инженер Самарского национального исследовательского университета имени академика С. П. Королева.

Daniil Andreevich SHISHKIN – first-year master's student in Aircraft Engineering and Engineer at Samara National Research University.

email: Cr1stra61@yandex.ru

ORCID: 0009-0008-9742-4489

УСМАНОВ Богдан Сергеевич – студент 1 курса магистратуры и инженер-конструктор Самарского национального исследовательского университета имени академика С.П. Королева. Специализируется в области авиастроения. Его научные интересы включают авиастроение, техническое зрение, AI-технологии и аэродинамические исследования микрорельефов.

Bogdan Sergeevich USMANOV – is a master's student and an engineer at Samara National Research University (also known as Samara University). He specializes in Aerospace Engineering. His research interests include aerospace engineering, Computer Vision, AI technologies, and aerodynamic studies of microreliefs.

email: usmanov.studios@yandex.ru

ORCID: 0009-0008-5078-7266

БОРГЕСТ Николай Михайлович – 1954 г. рождения. Окончил Куйбышевский авиационный институт им. академика С. П. Королева (1978), к. т. н. (1985). Доцент кафедры конструкции и проектирования летательных аппаратов Самарского национального исследовательского университета имени академика С. П. Королева. Член Международной ассоциации по онтологиям и их приложениям, Российской ассоциации искусственного интеллекта. В списке научных трудов более 200 работ в области автоматизации проектирования и ИИ.

Nikolay Mikhailovich BORGEST (b. 1954) graduated from the Kuibyshev Aviation Institute named after academician S.P. Korolev (Kuibyshev) in 1978, PhD (1985). He is an associate professor at the Samara National Research University. He is a member of the International Association for Ontology and its Applications, a member of the Russian Association of Artificial Intelligence, a co-author of more than 200 scientific articles and abstracts in the field of CAD and AI.

email: borgest@yandex.ru

ORCID: 0000-0003-2934-6198

Материал поступил в редакцию 13 марта 2026 года

УДК 004.81

КОГНИТИВНАЯ МОДЕЛЬ УПРАВЛЕНИЯ ТЕРМОЭЛЕМЕНТОМ ПЕЛЬТЬЕ

М. В. Бобырь[1] [0000-0002-5400-6817], А. А. Асеев[2] [0009-0007-8271-7660]

[1, 2]*Юго-Западный государственный университет, г. Курск, Россия*

[1]maxbobyr@gmail.com, [2]asseeff.artem@gmail.com

Аннотация

Представлена онтологическая модель системы управления термоэлементом Пельтье. Онтология описывает ее состав, выделяя объекты, процессы преобразования в объектах и атрибуты связей между ними. На основе разработанной онтологической модели спроектирована каскадная система управления, объединяющая ПИД-регулятор, нечетко-цифровой фильтр и экспоненциально усредняющий фильтр, причем ее когнитивное поведение основано на правилах нечеткой логики. Улучшение динамических характеристик переходных процессов системы управления термоэлементом Пельтье достигается за счет применения модели математических и онтологических решений, при этом каскадная система управления обеспечивает снижение амплитуды первой гармоники управляющего сигнала на 12% и сокращает время переходного процесса на 31.9%.

Ключевые слова*: онтология, нечеткая логика, ПИД-регулятор, нечетко-цифровой фильтр, экспоненциально усредняющий фильтр.*

ВВЕДЕНИЕ

Сложные технические системы в процессе проектирования требуют формализованного описания структуры системы и взаимодействия ее элементов. При традиционном подходе функционирование системы часто представляется в виде структуры «черный ящик», что не позволяет анализировать причинно-следственные связи обработки сигналов в ней. Это приводит к формированию слабо формализованных решений, затрудняющих анализ, модификацию и масштабирование системы. Одним из подходов, направленных на устранение указанного ограничения, является когнитивное моделирование, позволяющее формализовать состав системы, типы объектов внутри нее, процессы и связи между ними. При этом онтология в системе управления (СУ) позволяет формировать

единую интерпретацию архитектуры и обеспечивать согласованность между физическими процессами, аппаратной реализацией и вычислительными алгоритмами [1, 2].

Отсутствие онтологии в задаче управления сложными системами порождает два недостатка. Во-первых, инженер вынужден мысленно отслеживать всю цепочку процессов преобразования сигналов, что без явного описания и формализации может стать источником ошибок. Во-вторых, любое изменение системы управления (например, замена датчика) требует глубокого понимания зависимостей внутри нее, превращая сопровождение в трудоемкий процесс.

Рассмотрим задачу синтеза онтологии на примере управления термоэлементом [3, 4]. Как правило, в таких системах управления для их регулировки используется ПИД-регулятор [5–7]. Однако классическая реализация ПИД-управления обладает рядом ограничений. Во-первых, это трудоемкость ручной настройки коэффициентов ПИД-регулятора: пропорционального, интегрального и дифференциального [5, 6]. Во-вторых, это значительный скачок амплитуды первой гармоники управляющего сигнала при ПИД-регулировании [8, 9], приводящий к повышенному износу силовых элементов. Третьим ограничением является длительное время переходного процесса. Для преодоления перечисленных ограничений используются различные подходы. Трудоемкость настройки коэффициентов ПИД-регуляторов упрощается методами нейро-нечеткой и генетической оптимизации [6, 11], а также самонастраивающимся нечетким ПИД-регулятором [10], где скачок амплитуды управляющего сигнала уменьшался за счет использования двойного экспоненциального усреднения [12], а также нечеткого блока для адаптивной регуляции [14, 15]. Время переходного процесса сокращалось с помощью предиктивного управления [13], оптимизирующего сигнал с явными ограничениями на его изменение. Однако ПИД-регулятор используется только для компенсации отклонения между текущим и заданным целевыми значениями, в то время как онтология позволяет выполнять анализ динамики всех сигналов системы, участвующих в процессе управления. В предлагаемом подходе осуществляется контроль прохождения всех сигналов в системе управления термоэлементом. Выделяются этапы съема термистором и аналого-цифро-

вого преобразования сигнала измеренной температуры с термоэлемента, вычисления разницы между измеренной и заданной температурами и анализа динамики сигнала ПИД-регулятора в предыдущий и текущий моменты времени. Это обеспечивает интерпретируемость каждого управляющего решения, являющуюся основой объясняемого искусственного интеллекта. Система не только приводит управляемую величину к заданному значению, но и адаптирует динамику сигнала по явно сформулированным правилам, доступным для анализа и модификации. Ранее в работе [17] было предложено решение, в котором отсутствовало онтологическое описание системы. В настоящем исследовании указанный недостаток устранен путем описания всех вышеперечисленных процедур, использующихся в каскадной системе управления, объединяющей ПИД-регулятор, нечетко-цифровой фильтр (НЦФ) и экспоненциально усредняющий фильтр (ЭУФ) с адаптивным коэффициентом сглаживания. Разработанная онтологическая модель формализует состав системы, типы сигналов, вычислительные и физические процессы, а также причинно-следственные связи между ними. Такой подход позволяет рассматривать математическую модель и алгоритмы управления как прямое следствие формализованной онтологической структуры, обеспечивая интерпретируемость, воспроизводимость и ее масштабируемость.

ОНТОЛОГИЧЕСКАЯ МОДЕЛЬ СИСТЕМЫ УПРАВЛЕНИЯ ТЕРМОЭЛЕМЕНТОМ

Онтологическая модель системы управления термоэлементом (ОМСУТ) организует структуру системы управления и описывает взаимодействие между ее компонентами. ОМСУТ задается в виде следующего кортежа:

$$O = \langle O\text{о}_{i=1}^{a}, O\text{о}_{i=1}^{d}, O\text{п}_{i=1}^{p}, O\text{с}_{i=1}^{s} \rangle, \quad (1)$$

где $O\text{о}$ – онтологическая модель классов объектов (*a* = 1÷6, *a* – количество аналоговых объектов; *d* = 1÷8, *d* – количество цифровых объектов); $O\text{п}$ – онтологическая модель процесса (*p* = 1÷16, *p* – количество процессов); $O\text{с}$ – онтологическая модель сигналов (*s* = 1÷25, *s* – количество сигналов).

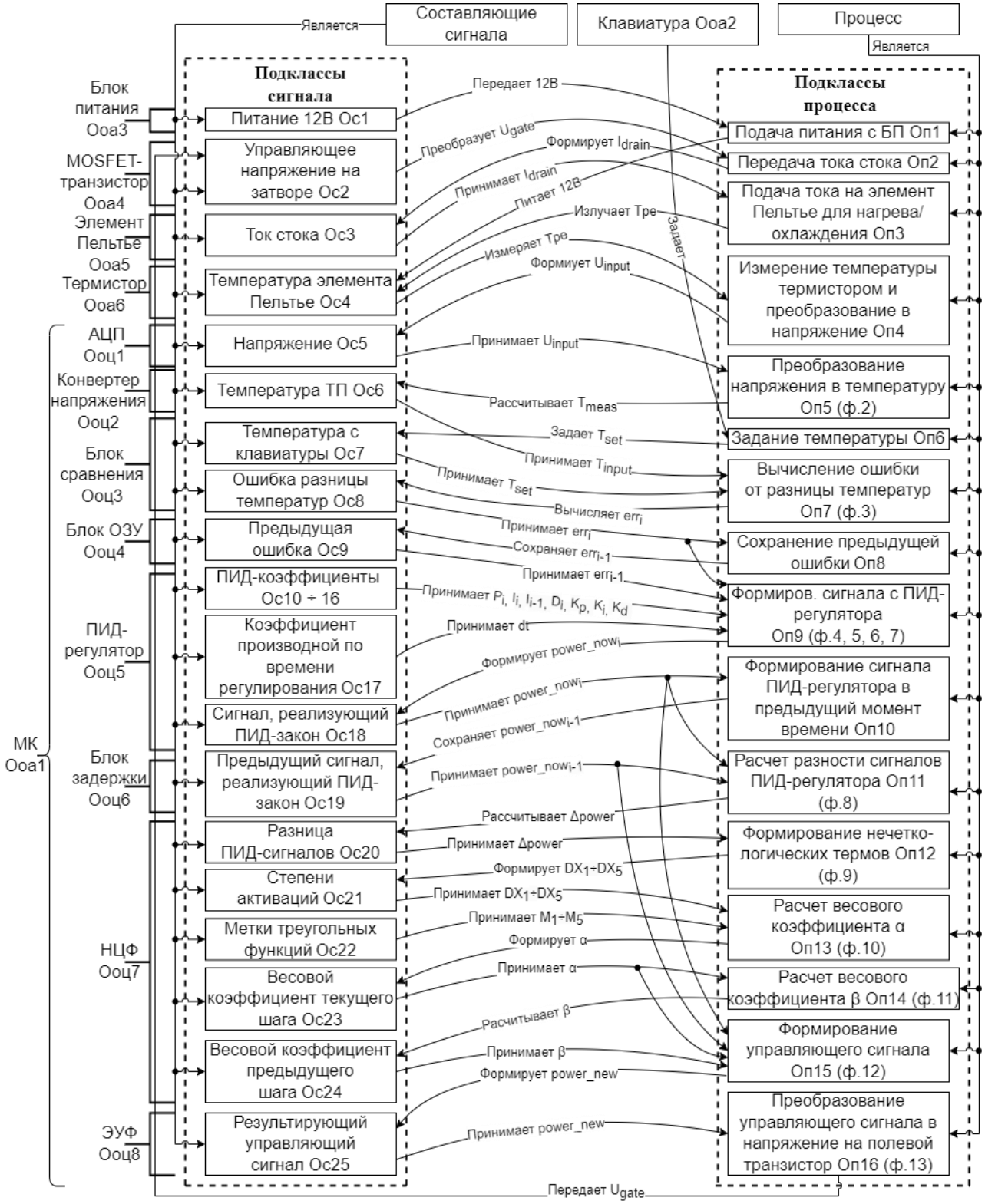


Рис. 1. Онтологическая модель каскадного управления термоэлементом.

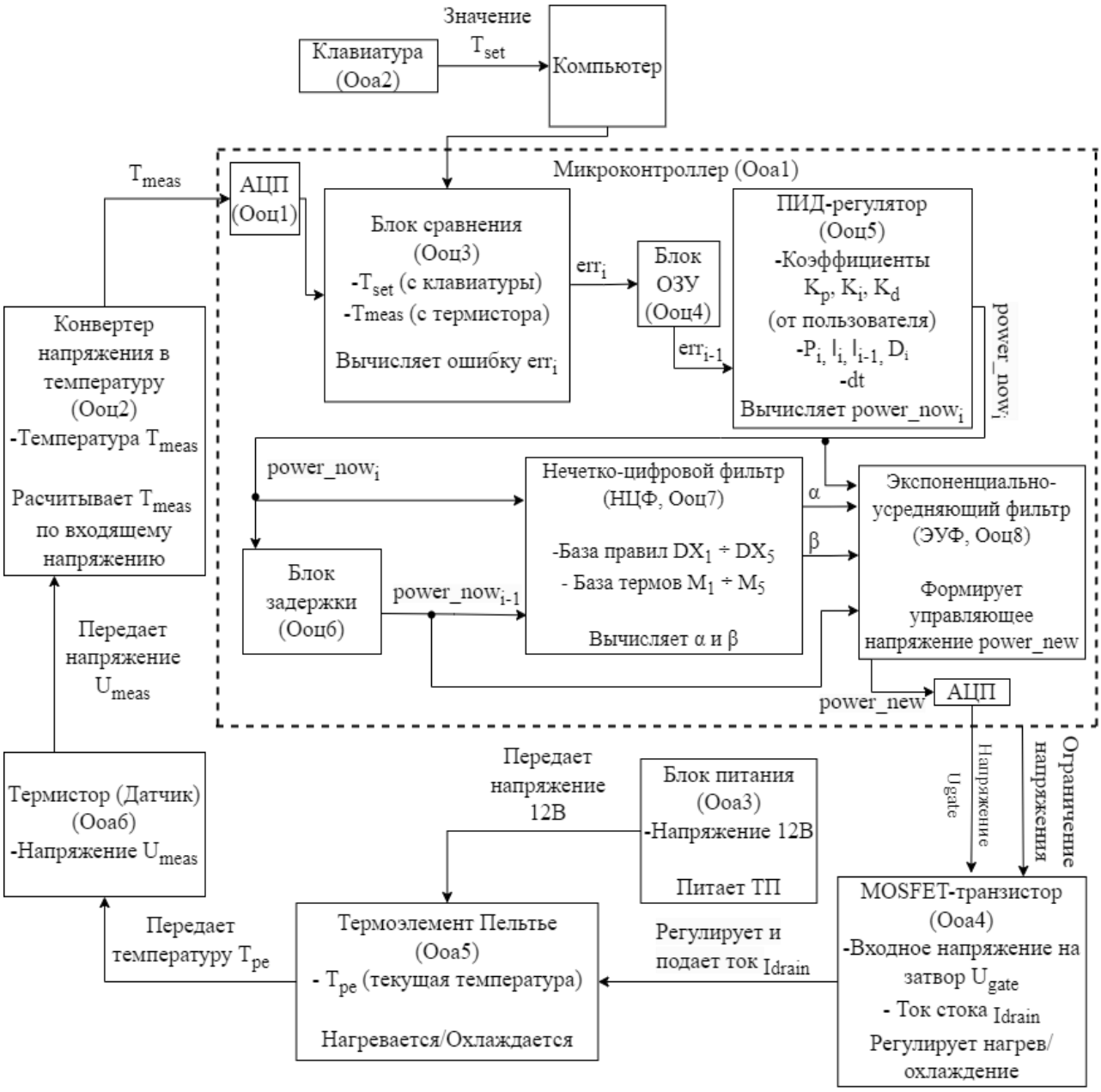


Рис. 2. Структурная схема каскадного управления термоэлементом

Модель определяет состав объектов системы, сигналов и процессы их преобразования (рис. 1), а также связи между ними (рис. 2). В рамках онтологической модели выделяются как физические объекты (термоэлемент Пельтье, термистор, MOSFET-транзистор), так и цифровые вычислительные блоки, а также логические и алгоритмические процессы, протекающие в микроконтроллере.

На уровне аналоговых объектов формируются электрические сигналы, соответствующие реальным физическим величинам, относящимся к температуре,

току или сопротивлению. Эти сигналы существуют непосредственно в физической среде и характеризуют состояние температуры на термоэлементе Пельтье (ТП) и связанных с ним компонентов. Датчики преобразуют физические величины в аналоговые электрические сигналы, пропорциональные измеряемым параметрам. Такие сигналы доступны для регистрации аппаратными средствами и подаются на вход аналого-цифрового преобразователя внутри микроконтроллера (МК), где их аналоговые значения переходят в цифровые. Значения цифровых сигналов используются в алгоритмах управления, программно реализованных в МК. На основе этих данных выполняются вычисления, включая определение ошибки регулирования и формирование управляющего воздействия. Результатом вычислений является электрический управляющий сигнал, подаваемый на исполнительные элементы системы, в частности на затвор MOSFET-транзистора, который формирует регулирующий сигнал тока стока и передает его на элемент Пельтье. Обратная связь в системе управления осуществляется путем определения ошибки регулирования между заданной температурой (T_{set}) и измеренной с поверхности ТП (T_{meas}).

Таким образом, онтология процессов управления включает задание целевой температуры, вычисление ошибки регулирования, формирование сигнала ПИД-регулятора, адаптивную фильтрацию ПИД-сигнала методами нечеткой логики, управление напряжением на затворе MOSFET-транзистора и обратную связь по температуре (рис. 2).

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ СИСТЕМЫ УПРАВЛЕНИЯ ТЕРМОЭЛЕМЕНТОМ

Математическая модель системы управления формируется на основе онтологической модели ОМСУТ и соответствует описанным в ней процессам и сигналам. Каждый математический алгоритм реализует конкретный онтологический процесс и использует сигналы, источники которых заданы в онтологии (рис. 1).

Формирование цифрового сигнала температуры

Аналоговая часть системы включает блок питания, который подает постоянное напряжение 12 В на ТП. Управление нагревом/охлаждением ТП осуществляется током стока (I_{drain}) MOSFET-транзистора, где управляющее напряжение

на затворе (U_{gate}) формируется каскадной системой (включающей ПИД-регулятор, НЦФ и ЭУФ), в зависимости от температуры на термоэлементе (T_{pe}).

Температура на термоэлемент измеряется с помощью термистора. Аналоговый сигнал напряжения (U_{meas}) от термистора передается в аналого-цифровой преобразователь МК, где по формуле

$$T_{\text{meas}} = 1.148\left(U_{\text{meas}}\ \frac{5}{1024}\right)^2 - 28.45(U_{\text{meas}}\ \frac{5}{1024}) + 87.85,$$

где 5 – значение опорного напряжения АЦП, 1024 – разрядность АЦП (10-разрядный), осуществляется преобразование сигнала в цифровое значение температуры (T_{meas}), которое в дальнейшем используется для управления ПИД-регулятором:

Коэффициенты полиноминальной регрессии был подобраны экспериментальным путем. В рамках эксперимента изменялась температура на термоэлементе и измерялась независимым прибором, в нашем эксперименте использовался тепловизор. Затем значения температуры фиксировались и сопоставлялись со значениями аналогового напряжения, поступающими от терморезистора. Далее на основе однофакторной регрессии находились коэффициенты полиномианального уравнения (2), которые составили 1.148, 28.45, 87.85 соответственно.

Реализация ПИД-закона

ПИД-регулятор стремится минимизировать ошибку регулирования err_i, определяемую блоком сравнения (рис. 1), как разность между заданной температурой (T_{set}) и измеренной температурой терморезистором (T_{meas}) по формуле

$$\text{err}_\text{i} = T_{\text{set}} - T_{\text{meas}}.$$

Формирование сигнала ПИД-закона осуществляется на основе трех составляющих: пропорциональной, интегральной и дифференциальной. Пропорциональная составляющая (P_i) определяется текущей величиной ошибки регулирования (err_i) по формуле

$$P_i = \text{err}_\text{i}.$$

Интегральная составляющая (I_i) рассчитывается по методу прямоугольников (метод Эйлера)

$$I_i = I_{i-1} + \mathrm{err_i} \cdot dt,$$

где I_{i-1} – значение интегральной составляющей, накопленное на предыдущем шаге дискретизации; *dt* – период дискретизации системы.

Для исключения насыщения интегратора вводится ограничение $I_{\min} \le I_i \le I_{\max}$.

Дифференциальная составляющая (D_i) аппроксимировалась обратной разностью по формуле

$$D_i = \frac{\mathrm{err}_i - \mathrm{err}_{i-1}}{dt},$$

где err_{i-1} – значение ошибки регулирования на предыдущем шаге дискретизации.

После вычисления отдельных составляющих формируется итоговый управляющий сигнал ПИД-регулятора. Сигнал, реализующий ПИД-закон ($\mathrm{power\_now}_i$) на *i*-м шаге дискретизации, вычисляется по формуле

$$\mathrm{power\_now}_i = K_p P_i + K_i I_i + K_d D_i,$$

где K_p, K_i, K_d – настраиваемые коэффициенты ПИД-регулятора.

Стоит отметить, что на первом шаге (*i* = 0) инициализация и хранение данных происходят следующим образом: принимается err_{i-1} = 0, I_{i-1} = 0. После каждого вычисления значение err_i сохраняется в переменную err_{i-1} (с использованием блока ОЗУ) для передачи этого сигнала на следующем шаге управления. Аналогично значение $\mathrm{power\_now}_i$ сохраняется как $\mathrm{power\_now}_{i-1}$ (с использованием блока задержки) для работы НЦФ и ЭУФ.

Нечетко-цифровая фильтрация

НЦФ служит для анализа изменения сигнала ПИД-регулятора и адаптивного расчета весового коэффициента *α* для последующего экспоненциально-усредняющего фильтрования.

Входной переменной НЦФ является разность между текущим и предыдущим значениями выходного сигнала ПИД-регулятора ($\Delta\mathrm{power}$). Она рассчитывается по формуле

$$\Delta\text{power} = \text{power\_now}_i - \text{power}_{\text{now}_{i-1}},$$

соответствующей онтологическому процессу, выполняемым НЦФ.

Величина Δpower характеризует «резкость» изменения сигнала ПИД-регулятора, содержит информацию о его направлении и описывается пятью треугольными функциями принадлежности (рис. 3), причем DX_1 и DX_5 имеют значения в виде трапеций, а DX_2, DX_3 и DX_4 – в виде треугольников. Таким образом, последняя формула реализует базовую когнитивную операцию, выделяющую динамику поведения Δpower, которая далее используется для принятия управляющих решений в рамках нечетко-логических правил. Для формирования базы нечетко-логических правил Δpower конвертируется в значения степеней активаций ($DX_1 \div DX_5$) по формулам

$$DX_1 = \begin{cases} 1, & \text{если } \Delta\text{power} \in [0; 1.2], \\ \dfrac{2.4 - \Delta\text{power}}{2.4 - 1.2}, & \text{если } \Delta\text{power} \in [1.2; 2.4], \\ 0, & \text{если } \Delta\text{power} \in [-\infty; 0] \cup [2.4; +\infty], \end{cases}$$

$$DX_2 = \begin{cases} \dfrac{\Delta\text{power} - 1.2}{2.4 - 1.2}, & \text{если } \Delta\text{power} \in [1.2; 2.4], \\ \dfrac{3.6 - \Delta\text{power}}{3.6 - 2.4}, & \text{если } \Delta\text{power} \in [2.4; 3.6], \\ 0, & \text{если } \Delta\text{power} \in [-\infty; 1.2] \cup [3.6; +\infty], \end{cases}$$

$$DX_3 = \begin{cases} \dfrac{\Delta\text{power} - 2.4}{3.6 - 2.4}, & \text{если } \Delta\text{power} \in [2.4; 3.6], \\ \dfrac{4.8 - \Delta\text{power}}{4.8 - 3.6}, & \text{если } \Delta\text{power} \in [3.6; 4.8], \\ 0, & \text{если } \Delta\text{power} \in [-\infty; 2.4] \cup [4.8; +\infty], \end{cases}$$

$$DX_4 = \begin{cases} \dfrac{\Delta\text{power} - 3.6}{4.8 - 3.6}, & \text{если } \Delta\text{power} \in [3.6; 4.8], \\ \dfrac{6.0 - \Delta\text{power}}{6.0 - 4.8}, & \text{если } \Delta\text{power} \in [4.8; 6.0], \\ 0, & \text{если } \Delta\text{power} \in [-\infty; 3.6] \cup [6.0; +\infty], \end{cases}$$

$$DX_5 = \begin{cases} 0, & \text{если } \Delta\text{power} \in [-\infty; 4.8] \cup [7.2; +\infty], \\ \dfrac{\Delta\text{power} - 4.8}{6.0 - 4.8}, & \text{если } \Delta\text{power} \in [4.8; 6.0], \\ 1, & \text{если } \Delta\text{power} \in [6.0; 7.2]. \end{cases}$$

Полученным степеням активаций функций принадлежности соответствуют пять нечетко-логических правил:

- *Если DX = DX*$_1$, то *α = DX*$_1$.
- *Если DX = DX*$_2$, то *α = DX*$_2$.
- *Если DX = DX*$_3$, то *α = DX*$_3$.
- *Если DX = DX*$_4$, то *α = DX*$_4$.
- *Если DX = DX*$_5$, то *α = DX*$_5$.

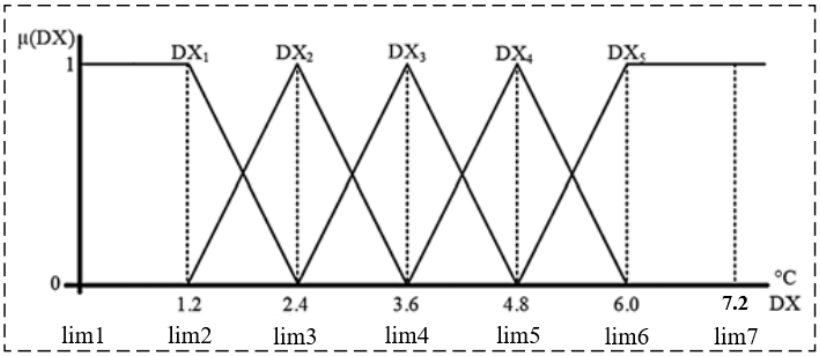


Рис. 3. Входные функции принадлежности переменной Δpower

Выходная лингвистическая переменная коэффициента сглаживания (*α*) также имеет пять термов с четкими значениями (синглтонами): *M*$_1$ = 0.4, *M*$_2$ = 0.5, *M*$_3$ = 0.6, *M*$_4$ = 0.7, *M*$_5$ = 0.8 (рис. 4).

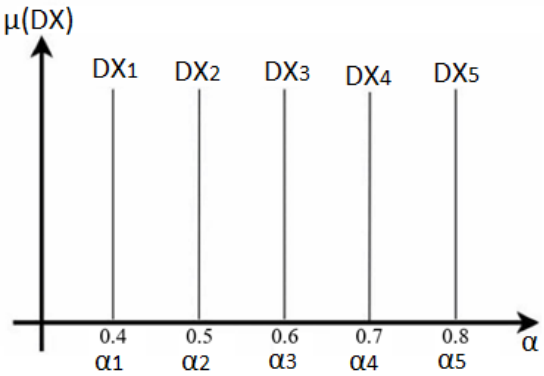


Рис. 4. Выходные синглтонные значения функций принадлежностей.

Для дефаззификации четкого численного значения коэффициента (α) применен упрощенный метод центра тяжести:

$$\alpha = \frac{\sum_{i=1}^{5} DX_i \cdot \alpha_i}{\sum_{i=1}^{5} DX_i}.$$

Расчет весового коэффициента (β) для ЭУФ выполнен по формуле

$$\beta = 1 - \alpha.$$

Экспоненциально усредняющая фильтрация

Экспоненциально-усредняющая фильтрация (ЭУФ) выполняет финальное адаптивное сглаживание передаваемого из НЦФ и ПИД-регулятора на затвор полевого транзистора управляющего сигнала ($power\_new$) по формуле

$$\text{power}_{\text{new}} = \alpha \text{power\_now}_i + \beta \text{power\_now}_{i-1}.$$

В отличие от классического экспоненциально-усредняющего фильтра с фиксированным коэффициентом сглаживания, коэффициент α динамически изменяется в диапазоне [0.4, 0.8], а β – в диапазоне [0.2, 0.6].

При резких изменениях Δpower коэффициент α становится малым (приближается к 0.4), что подавляет высокочастотные выбросы; при плавных изменениях α близок к 0.8, что обеспечивает быстрый отклик системы.

По вышеописанному принципу каскадная связка ПИД-регулятора с НЦФ + ЭУФ реализует фильтр нижних частот с адаптивной полосой пропускания, устраняя недостатки классического ПИД.

Формирование управляющего воздействия

Формула

$$U_{\text{gate}} = \text{power\_new} \cdot 2.5,$$

где 2.5 – масштабирующий коэффициент, соответствует онтологическому процессу передачи управляющего аналогового сигнала на полевой транзистор и обеспечивает переход от цифрового сигнала power_new к аналоговому сигналу U_{gate}. Процесс осуществляется в ЦАП микроконтроллера. Таким образом регулируется напряжение на затворе MOSFET-транзистора (U_{gate}), которое изменяет его ток

стока (I_{drain}). Ток стока полевого транзистора в свою очередь приводит к изменению тока на термоэлементе Пельтье и, как следствие, ведет к изменению его температуры (T_{pe}). Затем температура измеряется термистором и передается в микроконтроллер через АЦП (T_{meas}). По такому принципу образуется обратная связь. Описанные выше онтологическая и математическая модели формируют когнитивную модель управления термоэлементом.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ КАСКАДНОЙ МОДЕЛИ УПРАВЛЕНИЯ ТЕРМОЭЛЕМЕНТОМ

Для верификации предложенной модели и оценки ее эффективности по схеме, представленной на рис. 1, была собрана экспериментальная установка. В программном обеспечении микроконтроллера реализованы все процессы, описанные в разделе «Математическая модель управления термоэлементом». Период дискретизации системы (*dt*), рассчитанный суммой всех задержек внутри нее, установлен равным 0.6 с. В качестве целевой температуры выбрано значение T_{set} = 45˚C. Эксперимент состоял в сравнении показателей работоспособности двух систем управления: с классическим ПИД-регулятором (1) и каскадной системой на основе комбинации ПИД+НЦФ+ЭУФ:

1. *Система с классическим ПИД-регулятором*. Коэффициенты ПИД-регулятора: K_p = 2.0, K_i = 0.05, K_d = 1.0, *dt* = 0.01.

2. *Каскадная система ПИД+НЦФ+ЭУФ.* Параметры ПИД-регулятора оставались неизменными. Параметры НЦФ (границы функций принадлежности) установлены, как описано в п. «Нечетко-цифровая фильтрация».

В процессе измерялись и контролировались значения трех параметров: совпадение измеренной (T_{meas}) и заданной (T_{set}) температур, выброс амплитуды первой гармоники управляющего сигнала и время переходного процесса (U_1), т. е. момент времени, когда измеренная температура совпадет с заданной $T_{\text{meas}} = T_{\text{set}}$.

Результаты, полученные при классическом управлении ПИД-регулятором, представлены на рис. рис. 5а. При подаче управляющего напряжения на затвор MOSFET-транзистора (сигнал power_new) наблюдается ярко выраженный скачок напряжения в момент начала регулирования. Максимальная амплитуда U_1 этого скачка ограничена АЦП 100 усл. ед. (в действительности может быть и больше).

Временная ось оцифрована в отсчетах с периодом 0.6 с. Момент, когда сигнал измеренной температуры T_{meas} становится равным и остается в зоне допустимого отклонения от сигнала заданной температуры T_{set}, соответствует 361-му отсчету. Таким образом, время регулирования $T_{\text{B\_PID}}$ вычисляется по формуле

$$T_{\text{B\_PID}} = (N-1)\cdot dt = (361-1)\cdot 0.6 = 216 \text{ с.}$$

Значение напряжения управляющего сигнала U_1 при опорном напряжении, равном 5 В определяется по формуле

$$U_1 = 5.0 \cdot \frac{100}{100} = 5.0 \text{ В.}$$

Результаты, полученные при каскадном управлении с комбинацией ПИД + НЦФ + ЭУФ, представлены на рис. 5б. При подаче управляющего напряжения на затвор MOSFET-транзистора (сигнал power_new) демонстрируется значительно более плавный скачок напряжения U_1 по сравнению с классическим управлением при помощью ПИД-регулятора, амплитуда U_1 не превышает 88 усл. ед. Выход температуры (T_{meas}) на заданный режим происходит на 246-м отсчете. Время регулирования (T_B) вычисляется по формуле

$$T_{\text{B}} = (246-1)\cdot 0.6 = 147 \text{ с.}$$

Значению напряжения управляющего сигнала U_1 соответствует значение, рассчитанное по формуле

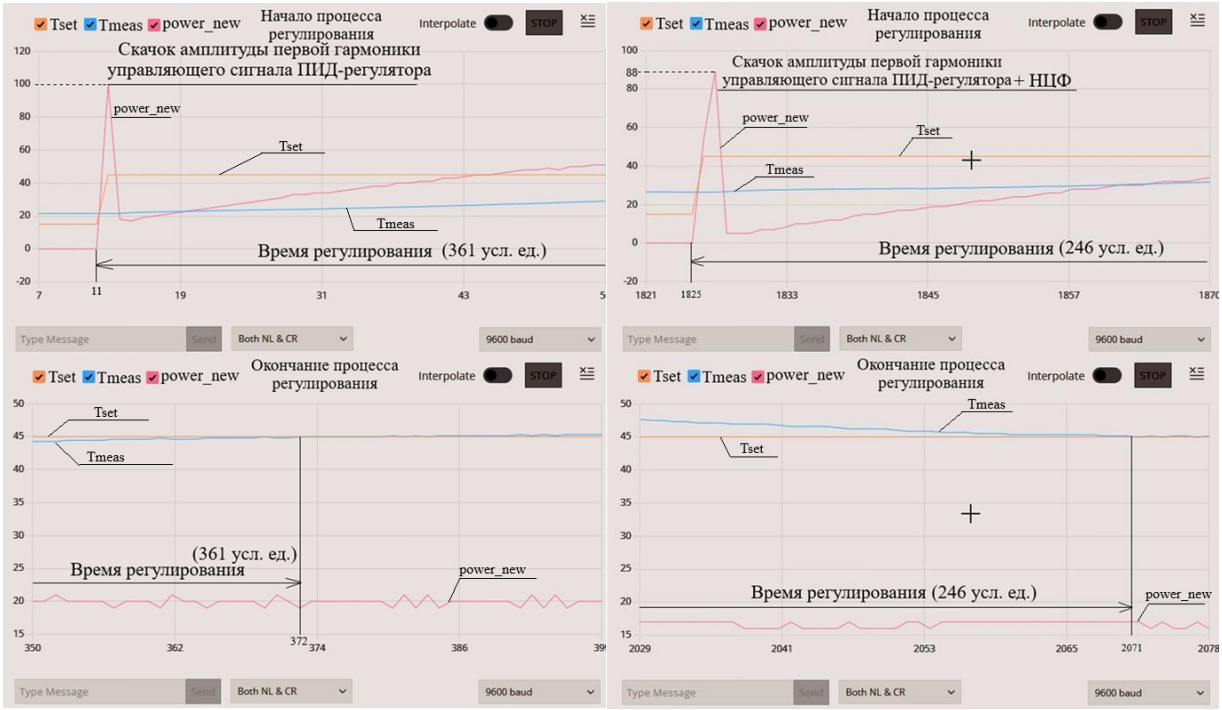
$$U_1 = 5.0 \cdot \frac{88}{100} = 4.4 \text{ В.}$$

Основные показатели процесса управления приведены в табл. 1.

Табл. 1: Сравнительные характеристики систем управления

Параметр	Классический ПИД	ПИД + НЦФ + ЭУФ	Абсолютное изменение	Относительное улучшение, %
Время регулирования, T_B, с	216	147	–69	31.9
Амплитуда первого скачка $\mathrm{power\_new}$, у.е.	100	88	–12	12.0
Усредненное напряжение скачка, U_1, В	5.0	4.4	–0.6	12.0

Таким образом, результаты эксперимента подтверждают преимущество каскадной системы управления ПИД + НЦФ + ЭУФ, состоящее в сокращении времени регулирования и снижении амплитуды управляющего сигнала.



а) б)

Рис. 5. Процессе управления термоэлементом: а) на основе классического ПИД-регулятора, б) каскадная система ПИД + НЦФ + ЭУФ

ЗАКЛЮЧЕНИЕ

Разработана когнитивная модель, включающая онтологическую и математическую модели управления термоэлементом Пельтье. Онтологическая модель формализует структуру системы, типы сигналов и процессы, а математическая модель реализует вычислительную интерпретацию этих процессов. Предложенная каскадная система управления, объединяющая ПИД-регулятор, нечетко-цифровой фильтр и экспоненциально усредняющий фильтр, обеспечивает адаптивную обработку управляющего сигнала в зависимости от его динамики. Экспериментальные результаты подтвердили эффективность представленной системы, поскольку время переходного процесса сократилось на 31.9%, а амплитуда управляющего сигнала снизилась на 12% по сравнению с классическим ПИД-регулятором.

Благодарности

Работа выполнена при поддержке Министерства науки и высшего образования в рамках выполнения работ по Государственному заданию № 075-03-2026-489.

СПИСОК ЛИТЕРАТУРЫ

1. *Grebeshkov A., Shebalov R., Gorshkov S., Mushtak O.* Ontological modelling of enterprises: technologies and methods // Book, TriniData LLC & Ural Federal University. 2019. P. 104–116. ISBN: 978-5-7996-2580-1.

2. *Krieken E., Acar E., van Harmelen F.* Analyzing Differentiable Fuzzy Logic Operators // Artificial Intelligence. 2022. Vol. 302, 103602. https://doi.org/10.1016/j.artint.2021.103602

3. *Jalahi A., Linke M., Weltzien C., Mahajan P.* Developing an Arduino-based control system for temperature-dependent gas modification in a fruit storage container // Computers and Electronics in Agriculture. 2022. Vol. 198, 107126. https://doi.org/10.1016/j.compag.2022.107126

4. *Loche-Moinet F., Theolier L., Woirgard E.* Electro-thermo-mechanical modelling of a SiC MOSFET transistor under non-destructive short-circuit // Microelectronics Reliability. 2023. Vol. 150, 115143. https://doi.org/10.1016/j.microrel.2023.115143

5. *Leva A., Zamuner M.* Model Parametrisation and Rule Selection for Problem-tailored PID Autotuning // IFAC-PapersOnLine. 2024. Vol. 58, Issue 7. P. 43–48. https://doi.org/10.1016/j.ifacol.2024.08.008

6. *Bassi S.J., Gbenga E.D., Abidemi A., Oyewola D., Mohammed B.K.* Metaheuristic Algorithms for PID Controller Parameters Tuning: Review, Approaches and Open Problems // Heliyon. 2022. Vol. 8, Issue 5, e09399. https://doi.org/10.1016/j.heliyon.2022.e09399.

7. *Keviczky L., Bányász C.* Adaptive Iterative Method to Improve the Robustness of PID Regulators // IFAC-PapersOnLine. 2022. Vol. 55, Issue 12. P. 149–155. https://doi.org/10.1016/j.ifacol.2022.07.303

8. Signe R.K., Motto F.B. Fuzzy-PID controller based sliding-mode for suppressing low frequency oscillations of the synchronous generator // Heliyon. 2024. Vol. 10, Issue 15, e35035. https://doi.org/10.1016/j.heliyon.2024.e35035

9. *Xian W., Qi Q., Liu W., Liu Y., Li D., Wang Y.* Control of quadrotor robot via optimized nonlinear type-2 fuzzy fractional PID with fractional filter: Theory and experiment // Aerospace Science and Technology. 2024. Vol. 151, 109286. https://doi.org/10.1016/j.ast.2024.109286.

10. *Outanoute M., Selmani A., Oubehar H., Snoussi A., Guerbaoui M., Ed-Dahhak A., Lachhab A., Bouchikhi B.* Self Tuning Fuzzy-PID Controller in Real Time Greenhouse Temperature Control // Conference: The Third International Conference on Optimization and Applications (ICOA 2017) At: Meknes, Morocco; 2017.

11. *Bobyr M.V., Milostnaya N.A., Bulatnikov V.A.* The fuzzy filter based on the method of areas' ratio // Applied Soft Computing. 2022. Vol. 117, 108449. https://doi.org/10.1016/j.asoc.2022.108449

12. *Ma Z., Pan T., Tian J.* Deep reinforcement learning optimized double exponentially weighted moving average controller for chemical mechanical polishing processes // Chemical Engineering Research and Design. 2023. Vol. 197. P. 419–433. https://doi.org/10.1016/j.cherd.2023.07.049

13. *Bobyr M., Titov V., Belyaev A.* Fuzzy System of Distribution of Braking Forces on the Engines of a Mobile Robot // MATEC Web of Conferences. 2016. Vol. 79, 01052. https://doi.org/10.1051/matecconf/20167901052

14. *Barelli L., Bidini G., Arce R.* Fuzzy Logic Regulator for the Performance Improvement and the Energy Consumption Reduction of an Industrial Chiller // Conference: ASME 2003 International Mechanical Engineering Congress and Exposition, 2008, 41910. https://doi.org/10.1115/IMECE2003-41910

15. *Bobyr M., Arkhipov A., Emelyanov S., Milostnaya N.* A method for creating a depth map based on a three-level fuzzy model // Engineering Applications of Artificial Intelligence. 2023. Vol. 117, 105629. https://doi.org/10.1016/j.engappai.2022.105629

16. *Bobyr M.V., Arkhipov A.E., Yakushev A.S.* Shade recognition of the color label based on the fuzzy clustering // Informatics and Automation. 2021. Vol. 20 (2). P. 407–434. https://doi.org/10.15622/ia.2021.20.2.6

17. *Бобырь М.В., Милостная Н.А., Ноливос К.А.* Комбинация нечетко-цифрового фильтра и ПИД регулятора в задаче управления термоэлементом // Мехатроника, автоматизация, управление. 2022. Том 23. https://doi.org/ 10.17587/mau.23.473-480

COGNITIVE MODEL FOR CONTROL OF A PELTIER THERMOELEMENT

M. V. Bobyr[1] [0000-0002-5400-6817], A. A. Aseev[2] [0009-0007-8271-7660]

[1, 2]*South-West State University, Kursk, Russia*

[1]maxbobyr@gmail.com, [2]asseeff.artem@gmail.com

Abstract

The article presents an ontological model of a control system for a Peltier thermoelectric element. The ontology describes the structure of the system by identifying objects, transformation processes within these objects, and the attributes of the relationships between them. Based on the developed ontological model, a cascade control system has been designed, integrating a PID controller, a fuzzy-digital filter, and an exponential-averaging filter, with its cognitive behavior governed by fuzzy logic rules. Improvement of the dynamic characteristics of transient processes in the Peltier element control system is achieved through the application of the mathematical and ontological solutions specified in the model. The cascade control system reduces the amplitude of the first harmonic of the control signal by 12% and decreases the transient response time by 31.9%.

Keywords*: ontology, fuzzy logic, PID-controller, fuzzy-digital filter, exponential-averaging filter.*

Acknowledgements

This work was supported by the Ministry of Science and Higher Education under State Contract No. 075-03-2026-489.

REFERENCES

1. *Grebeshkov A., Shebalov R., Gorshkov S., Mushtak O.* Ontological modelling of enterprises: technologies and methods // Book, TriniData LLC & Ural Federal University. 2019. P. 104–116. ISBN: 978-5-7996-2580-1.

2. *Krieken E., Acar E., van Harmelen F.* Analyzing Differentiable Fuzzy Logic Operators // Artificial Intelligence. 2022. Vol. 302, 103602. https://doi.org/10.1016/j.artint.2021.103602

3. *Jalahi A., Linke M., Weltzien C., Mahajan P.* Developing an Arduino-based control system for temperature-dependent gas modification in a fruit storage container // Computers and Electronics in Agriculture. 2022. Vol. 198, 107126. https://doi.org/10.1016/j.compag.2022.107126

4. *Loche-Moinet F., Theolier L., Woirgard E.* Electro-thermo-mechanical modelling of a SiC MOSFET transistor under non-destructive short-circuit // Microelectronics Reliability. 2023. Vol. 150, 115143. https://doi.org/10.1016/j.microrel.2023.115143

5. *Leva A., Zamuner M.* Model Parametrisation and Rule Selection for Problem-tailored PID Autotuning // IFAC-PapersOnLine. 2024. Vol. 58, Issue 7. P. 43–48. https://doi.org/10.1016/j.ifacol.2024.08.008

6. *Bassi S.J., Gbenga E.D., Abidemi A., Oyewola D., Mohammed B.K.* Metaheuristic Algorithms for PID Controller Parameters Tuning: Review, Approaches and Open Problems // Heliyon. 2022. Vol. 8, Issue 5, e09399. https://doi.org/10.1016/j.heliyon.2022.e09399.

7. *Keviczky L., Bányász C.* Adaptive Iterative Method to Improve the Robustness of PID Regulators // IFAC-PapersOnLine. 2022. Vol. 55, Issue 12. P. 149–155. https://doi.org/10.1016/j.ifacol.2022.07.303

8. Signe R.K., Motto F.B. Fuzzy-PID controller based sliding-mode for suppressing low frequency oscillations of the synchronous generator // Heliyon. 2024. Vol. 10, Issue 15, e35035. https://doi.org/10.1016/j.heliyon.2024.e35035

9. *Xian W., Qi Q., Liu W., Liu Y., Li D., Wang Y.* Control of quadrotor robot via optimized nonlinear type-2 fuzzy fractional PID with fractional filter: Theory and experiment // Aerospace Science and Technology. 2024. Vol. 151, 109286. https://doi.org/10.1016/j.ast.2024.109286.

10. *Outanoute M., Selmani A., Oubehar H., Snoussi A., Guerbaoui M., Ed-Dahhak A., Lachhab A., Bouchikhi B.* Self Tuning Fuzzy-PID Controller in Real Time Greenhouse Temperature Control // Conference: The Third International Conference on Optimization and Applications (ICOA 2017) At: Meknes, Morocco; 2017.

11. *Bobyr M.V., Milostnaya N.A., Bulatnikov V.A.* The fuzzy filter based on the method of areas' ratio // Applied Soft Computing. 2022. Vol. 117, 108449. https://doi.org/10.1016/j.asoc.2022.108449

12. *Ma Z., Pan T., Tian J.* Deep reinforcement learning optimized double exponentially weighted moving average controller for chemical mechanical polishing processes // Chemical Engineering Research and Design. 2023. Vol. 197. P. 419–433. https://doi.org/10.1016/j.cherd.2023.07.049

13. *Bobyr M., Titov V., Belyaev A.* Fuzzy System of Distribution of Braking Forces on the Engines of a Mobile Robot // MATEC Web of Conferences. 2016. Vol. 79, 01052. https://doi.org/10.1051/matecconf/20167901052

14. *Barelli L., Bidini G., Arce R.* Fuzzy Logic Regulator for the Performance Improvement and the Energy Consumption Reduction of an Industrial Chiller // Conference: ASME 2003 International Mechanical Engineering Congress and Exposition, 2008, 41910. https://doi.org/10.1115/IMECE2003-41910

15. *Bobyr M., Arkhipov A., Emelyanov S., Milostnaya N.* A method for creating a depth map based on a three-level fuzzy model // Engineering Applications of Artificial Intelligence. 2023. Vol. 117, 105629. https://doi.org/10.1016/j.engappai.2022.105629

16. *Bobyr M.V., Arkhipov A.E., Yakushev A.S.* Shade recognition of the color label based on the fuzzy clustering // Informatics and Automation. 2021. Vol. 20 (2). P. 407–434. https://doi.org/10.15622/ia.2021.20.2.6

17. *Bobyr M.V., Milostnaya N.A., Nolivos K.A.* Combination of a fuzzy-digital filter and a PID controller in the problem of controlling a thermoelement // Mechatronics, automation, control. 2022. Vol. 23. https://doi.org/ 10.17587/mau.23.473-480

СВЕДЕНИЯ ОБ АВТОРАХ

БОБЫРЬ Максим Владимирович – 1978 года рождения, учился в Курском государственном техническом университете (ныне – Юго-Западный государственный университет). В 2012 году защитил диссертацию на соискание доктора технических наук по специальности 05.13.06 «Автоматизация и управление технологическими процессами и производствами». В настоящее время работает профессором на кафедре программной инженерии. Является председателем диссертационного совета по специальности 5.12.4 «Когнитивное моделирование».

Область научных интересов: адаптивные нейро-нечеткие системы вывода, цифровая обработка изображений и распознавание образов, машинное обучение, стереозрение.

Maksim Vladimirovich BOBYR – born in 1978, studied at Kursk State Technical University (now South-West State University). In 2012, he defended his dissertation for the degree of Doctor of Technical Sciences in specialty 05.13.06 Automation and Control of Technological Processes and Production. Currently, he works as a professor at the Department of Software Engineering. He is the chairman of the dissertation council for specialty 5.12.4 Cognitive Modeling.

Research interests: adaptive neuro-fuzzy inference systems, digital image processing and pattern recognition, machine learning, stereo vision.

email: maxbobyr@gmail.com

ORCID: 0000-0002-5400-6817

АСЕЕВ Артем Андреевич – 1999 года рождения, учится на 3 курсе аспирантуры Юго-Западного государственного университета на кафедре программной инженерии по специальности 2.3.3 Автоматизация и управление технологическими процессами и производствами. Область научных интересов: адаптивные нейро-нечеткие системы вывода, распознавание образов.

Aseev Artem Andreevich – born in 1999, 3th-year postgraduate student at the Southwestern State University, Department of Software Engineering, specialty 2.3.3 Automation and Control of Technological Processes and Production. Research interests: adaptive neuro-fuzzy inference systems, pattern recognition.

email: asseeff.artem@gmail.com

ORCID: 0009-0007-8271-7660

Материал поступил в редакцию 25 марта 2026 года

УДК 004.42+37.01

АЛГОРИТМЫ ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ НА ОСНОВЕ КОМПОЗИЦИИ РЕЗУЛЬТАТОВ ПЕДАГОГИЧЕСКИХ ЭКСПЕРИМЕНТОВ

М. С. Дьяченко[0000-0002-5809-4981]

Национальный исследовательский центр «Курчатовский институт», г. Москва, Россия

mdyachenko@niisi.ru

Аннотация

Рассмотрены различные аспекты практической реализации алгоритмов индивидуализированного обучения (основанные на результатах педагогических экспериментов) при обучении с преподавателем (в аудитории, дистанционно или в гибридном режиме) и при самостоятельной работе студента. Описанная система одновременно обучает студента материалам курса и приемам самостоятельного обучения, то есть образовательным технологиям, которые формируют индивидуальную образовательную траекторию. Подмножество образовательных технологий определяется индивидуально для каждого студента в группе. Образовательные технологии независимы от учебного курса и универсальны, поэтому могут применяться на последующих или параллельных курсах. Преподаватели могут описывать новые образовательные технологии в виде скриптов на языке Python без привлечения разработчиков. Предложенная реализация интегрируется с цифровой образовательной платформой Мирера для расширения возможностей платформы.

***Ключевые слова**: индивидуализация обучения, автоматизированная система обучения, цифровая образовательная платформа, адаптивное обучение.*

ВВЕДЕНИЕ

Опережающее развитие экономики требует ускоренной подготовки большого числа высококвалифицированных кадров, что, в свою очередь, ставит задачу поиска подходов, позволяющих существенно увеличить темп, сроки и численность специалистов, подготавливаемых системой образования, без ущерба

качеству образования, а в ряде случаев даже с улучшением средних показателей компетентности выпускников. Так как на подготовку преподавателя вуза уходит не менее 5 лет, а текущие педагогические кадры уже работают на полную ставку, для увеличения количества подготавливаемых специалистов наиболее очевидным решением является увеличение размера учебных групп. Вместе с тем одним из главных требований для обучения больших групп в вузе выступает поддержание высокого уровня образовательных результатов с учетом ограниченных возможностей преподавателя вести персональную работу с каждым учащимся в условиях роста учебной нагрузки на преподавателя.

Для сохранения баланса между качеством обучения и размером учебной группы внедряют инструменты, облегчающие выполнение рутинных повторяющихся задач, например, таких как оценка заданий, выполненных студентами. Большинство современных решений в области автоматизации процесса обучения ориентированы на автоматизированную проверку заданий. Значительно меньшее число решений способно обеспечить адаптацию учебного процесса к индивидуальным особенностям каждого обучающегося, снимая эту задачу с преподавателя.

Адаптация учебных материалов, постоянно выполняемая преподавателем при работе с обучаемыми в группе, является важным фактором успешного освоения студентами учебного курса. Автоматизированные системы обучения берут на себя функцию преподавателя по адаптации учебных материалов как при смешанной форме обучения с преподавателем, так и при самостоятельной работе студента. Такие системы обучения должны способствовать развитию навыков самообразования, реализуемых традиционно при индивидуальной работе с участием преподавателя. Навыки самообразования позволяют студентам продуктивно усваивать материал как на занятиях, так и при самостоятельной работе с учебными материалами. Развитие у студентов навыков самообучения также способствует снижению нагрузки на преподавателя, поскольку студенты справляются с большей частью учебного материала самостоятельно, за счет чего у преподавателя высвобождаются ресурсы для работы со студентами, которым не могут помочь ограниченные по возможностям алгоритмы автоматизированной системы, поэтому требуется помощь преподавателя-человека.

ПОСТАНОВКА ПРОБЛЕМЫ

Существующие системы автоматизации обучения преимущественно реализуют выбранные разработчиками технологии адаптации учебных материалов, используемые для улучшения отдельных аспектов обучения. Однако применяемые технологии не предназначены для поиска индивидуальной образовательной технологии для каждого обучаемого в группе, а, как показывают исследования, эффективны только для обучаемых, восприимчивых к используемой образовательной технологии. Под образовательными технологиями в настоящей статье понимаются приемы обучения и небольшие по объему и простые по структуре отдельные шаги метода обучения, например повтор ранее изученного материала определенного уровня сложности, изучение нового материала для подготовки к занятию и т. д. Автоматизация образовательной технологии заключается в выборе следующего учебного действия на основании данных, собранных при взаимодействии студента с автоматизированной системой обучения.

Результаты эксперимента, полученные в исследовании [1], дали возможность установить, что при работе студентов в системе адаптивного обучения ALEKS в течение четырех месяцев замечено снижение навыков самообучения, которые должны были сохраняться и совершенствоваться в процессе обучения.

Как показывает выполненный обзор результатов исследований в области самообучения студентов [2], помимо демонстрации академической успеваемости для студентов становится необходимым развитие устойчивых навыков самообучения, которые исследователи отнесли к важным инструментам достижения академических успехов и поддержки послевузовского профессионального развития. Осознанная самообразовательная компетенция студентов, как отмечено в [3], является предпосылкой для формирования высококвалифицированного специалиста.

Цель настоящей работы – рассмотреть различные аспекты практической реализации алгоритмов индивидуализированного обучения (основанных на результатах педагогических экспериментов) при обучении с преподавателем (в аудитории, дистанционно или в гибридном режиме) и при самостоятельной работе студента.

ОБЗОР СУЩЕСТВУЮЩИХ ПОДХОДОВ

В России проведены успешные исследования в области использования систем адаптивного обучения. Наиболее масштабное исследование выполнено в Сибирском федеральном университете [4]. Менее масштабные эксперименты, также подтвердившие эффективность адаптивного обучения, осуществлены в Российском экономическом университете имени Г. В. Плеханова [5], Башкирском государственном аграрном университете [6], а также в Самарском государственном медицинском университете [7]. Исследование [8] охватило одновременно Московский государственный университет имени М. В. Ломоносова (МГУ), Санкт-Петербургский политехнический университет Петра Великого, Новосибирский национальный исследовательский государственный университет, Казанский (Приволжский) федеральный университет и Дальневосточный федеральный университет. Все отмеченные работы продемонстрировали эффективность адаптивного подхода, однако разработанные приемы не получили широкого распространения в системе образования, не в последнюю очередь, из-за специфичных условий применения, ограниченной возможностью переноса адаптивного алгоритма между курсами, ограничений по переносу технической реализации и необходимости трудоемкой разработки адаптивных учебных материалов.

Результаты исследований, выполненных в Томском государственном университете и Университете науки и технологий МИСИС, были коммерциализированы в отечественной системе адаптивного обучения Plario (компания ENBISYS, Томск), основанной на подходе Bayesian Knowledge Tracing. Система Plario внедрена в нескольких вузах. Для разработки обучающих курсов названная компания привлекает группы экспертов. По информации на сайте компании по состоянию на 2026 год реализованы единицы адаптивных курсов[1]. В исследовании [9], посвященном применению Plario для обучения математики на пером курсе, определено, что Plario – это прежде всего цифровой репетитор, предназначенный для выравнивания знаний, умений и навыков. В этом исследовании нет результатов оценки влияния системы на развитие компетенции самообучения, хотя от-

[1]Plario – система адаптивного обучения https://plario.ru/

мечено, что работа с системой предполагает увеличение объема самостоятельной работы и снижения количества аудиторных часов.

Адаптивное обучение также представлено в системе Stepik. Так, в проведенном в Казахстане исследовании [10] показано, что обучение математике на адаптивном курсе дает улучшение средней оценки, а также помогает развить навык самооценки знаний студента.

Кроме того, адаптивные технологии используются и на платформе Яндекс.Практикум, однако нам не удалось найти научные публикации об оценке результативности обучения на этой платформе, хотя есть связанные с этим исследования [11] о применении API большой языковой модели от Яндекса YandexGPT при обучении программированию.

Зарубежные образовательные учреждения также широко используют платформы адаптивного обучения. Согласно исследованию [12], в число таких платформ входят Knewton Alta[2], Smart Sparrow[3], DreamBox Learning[4] и ALEKS[5]. Активно развиваются системы адаптивного обучения в Китае. Так, система Squirrel AI[6] по состоянию на 2026 год используется десятками тысяч студентов. Согласно результатам [13], по итогам обучения не только вырос средний балл, но и явно отмечено улучшение результатов при отложенной оценке остаточного уровня знаний.

В адаптивной системе обучения учебный материал представлен набором модулей – учебных элементов, каждый из которых содержит пререквизиты для начала обучения, материал для изучения в виде текста, интерактивного элемента или видео, набор проверочных заданий различных уровней сложности или генератор заданий. Обучение проходит самостоятельно, изолированно или совместно с преподавателем. Сначала студенты осваивают содержание модуля, после чего проводится проверка усвоения материала. На основе результатов

[2]Knewton Alta – система адаптивного обучения https://www.wiley.com/en-ie/grow/teach-learn/teacher-resources/courseware/knewton-alta/

[3]Smart Sparrow – система адаптивного обучения https://www.smartsparrow.com/

[4]DreamBox Learning – система адаптивного обучения https://www.dreambox.com/

[5]ALEKS – система адаптивного обучения https://www.aleks.com/index.html

[6]Squirrel AI – система адаптивного обучения https://squirrelai.com/

проверки осуществляется адаптация индивидуальной образовательной траектории учащегося. В основном адаптация заключается в изменении порядка изучения тем, уровня сложности заданий либо уровня детализации и формы представления задания и результата проверки решения.

В исследовании [14] показано, что в автоматизированных системах обучения объектами адаптации являются содержание учебного материала, порядок его изучения и учебные задания. Кроме того, отмечено, что автоматизированные системы на практике не реализуют адаптацию сразу всех объектов, ограничиваясь одним или двумя.

Методика адаптации разнообразна. Так, в системе обучения немецкому языку [15] используются одновременно управление сложностью заданий и уровнем детализации, проактивные подсказки, информирование о статистике прохождения заданий и наиболее частых ошибках. Авторы [16] реализовали адаптацию порядка изучения материалов на основе кривых забывания Эббингауза с использованием генетических алгоритмов. Оригинальное исследование кривых забывания Эббингауза (1885 г.) было повторено в 2015 г., получено подтверждение основных результатов оригинального исследования [17]. Активно развиваются подходы, основанные на больших объемах накопленных данных. Например, авторы [18] используют глубокое обучение для адаптации траектории обучения. В последующих исследованиях продемонстрированы решения с глубоким обучением, в том числе с механикой забывания [19]. Однако на практике даже простые реализации на основе эвристик показывают эффективность. Так, в работе [4] эвристики использовались для управления сложностью заданий и выбора языка представления учебных материалов.

Перечисленные выше работы объединяет использование единого алгоритма адаптации или набора алгоритмов для всех студентов в группе, однако результаты экспериментов показывают, что не все студенты группы получают положительный эффект от применения единого параметрического алгоритма. Студенты, испытывающие сложности при обучении в системе, для которых оказалось недостаточно возможностей по адаптации, по условию эксперимента могли обращаться непосредственно к преподавателю для решения возникающих проблем.

Анализ самообучения в автоматизированных системах проводится реже, так как предполагается, что алгоритм автоматизированной системы находит индивидуальные параметры обучения студента и автоматически формирует траекторию обучения – студент получает от алгоритма готовый результат и сам в процессе выбора образовательной технологии не участвует.

В исследовании [20] самообучение стимулировалось за счет применения геймификации на уровне учебной группы. Процесс обучения, предложенный в этой работе, провоцировал конкуренцию внутри группы в борьбе за единый ресурс-оценку, что могло приводить к выбору студентом стратегии, предполагающей активную самостоятельную работу. Однако существенными ограничениями этого подхода были требование к наличию у студента мотивации к обучению, а также отсутствие контроля применения методов самообучения.

Анализ результатов, представленных в [8], показал, что после внедрения системы адаптивного обучения было зарегистрировано повышение уровня оценки компетенции «Самообучение», однако оценка проводилась на основании тестирования студентов экспериментальной и контрольной групп. Автор также отмечает развитие метакогнитивных навыков, например, в части планирования.

Несмотря на значительный прогресс в области внедрения адаптивного обучения в России, существующие подходы демонстрируют положительные эффекты лишь для части студентов, в том числе из-за использования универсальных параметрических алгоритмов, которые реализуются на уровне системы автоматизации обучения сразу для всех студентов в группе. При этом формирование универсальных алгоритмов персонализации, которые подходили бы всем студентам, остается нерешенным вопросом. Кроме того, недостаточно исследован вопрос самообучения в адаптивных системах. В связи с этим актуальной является задача реализации алгоритма индивидуализации обучения, который позволил бы комбинировать результаты проведенных педагогических экспериментов для поиска индивидуальных подходов для каждого студента. При этом подход, реализуемый алгоритмом, должен быть переносимым между различными курсами и должен стимулировать развитие навыков самообучения у обучаемых. Развитие навыков самообучения у студентов снижает нагрузку на преподавателя

и дает возможность увеличивать учебные группы, а также является важным инструментом формирования высококвалифицированного специалиста.

МЕТОДЫ ИССЛЕДОВАНИЯ

Результаты, представленные в настоящей статье, получены из основе анализа экспериментов, проведенных в России и за рубежом в области внедрения и эксплуатации систем адаптивного обучения, а также исследований по оценке уровня развития навыков самообучения у студентов при взаимодействии с автоматизированными системами обучения. Были рассмотрены технологии реализации адаптивности, подходы к разработке адаптивных материалов, а также характер взаимодействия обучаемых с учебной системой.

Отмеченные нами исследования показали, что фокус разработчиков обучающих систем находится только на обучении материалу курса, при этом были установлены краткосрочные цели обучения – добиться результатов в конкретном учебном курсе. В отдельных исследованиях оценивались уровень остаточных знаний или влияние процесса обучения на компетенции в области самообучения, но не делались выводы о способах улучшения компетенции самообучения.

Ниже рассмотрены базовые методы адаптивного обучения и такие методики, как скаффолдинг (skaffolding), которая изначально предполагает снижение участия преподавателя (предоставляющего поддержку «строительными лесами») – в нашем случае автоматизированной системы обучения – по мере освоения обучаемым предложенного материала. Применив методику скаффолдинг к процессу самообучения, мы пришли к выводу, что обучение навыку самообучения должно быть встроено в процесс обучения студента, следовательно, по мере обучения студента уровень вовлечения автоматизированной системы обучения должен снижаться.

Новыми сущностями в системе автоматизированного обучения будут прием обучения и учебный элемент, обладающие своими жизненными циклами. Система должна обеспечить реализацию жизненных циклов этих сущностей.

По результатам исследования синтезированы требования к системе автоматизированного обучения, описаны основные модели данных и разработан

прототип, подтверждающий возможность создания системы адаптивного обучения на базе разработанных принципов.

МЕТОД АВТОМАТИЧЕСКОГО ФОРМИРОВАНИЯ ИНДИВИДУАЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ ТРАЕКТОРИИ

Используемый подход заключается в автоматическом формировании индивидуальной образовательной траектории (ИОТ) обучения студента за счет применения композиции образовательных технологий (ОТ).

Образовательные технологии являются приемами обучения, которые студент может применять самостоятельно вне автоматизированной системы. ИОТ формируется в процессе самостоятельного обучения студента, когда в образовательном процессе не участвует человек-преподаватель. При формировании ИОТ система отбирает автоматизированные приемы обучения, применение которых демонстрирует результаты для конкретного студента. После отбора приемов и подбора значений их параметров для конкретного студента система стимулирует самостоятельное освоение им учебного материала и далее переходит к контролю его усвоения. Предлагаемый подход описан в [21].

Каждая образовательная технология имеет свой жизненный цикл, включающий этапы отбора технологии по эффективности, подбора значений параметров технологии, освоения и контроля владением ею. Поскольку приемы обучения являются результатами педагогической практики или педагогических экспериментов, они описаны в виде, применимом как для автоматической адаптации учебного материала, так и для автоматического контроля результата обучения. Дополнительно к адаптации материала и проверке уровня знаний система также реализует функцию обучения студента применению образовательных технологий, которые продемонстрировали свою эффективность (рис. 1).

На этапе оценки эффективности (рис. 1) система начинает использовать автоматизированную ОТ при формировании ИОТ и оценивает результаты применения ОТ. В случае подтверждения результативности ОТ система начинает обучение студента самостоятельному применению ОТ, направленному на выработку устойчивого навыка, постепенно снижая уровень своего влияния на студента, обучая его самостоятельному принятию решений. Для обучения навыку применения ОТ система начинает предлагать студенту возможные варианты

адаптации и ожидает, что студент выберет из них соответствующую ОТ. Если студент некорректно выбирает вариант адаптации, то система приводит описание ОТ с примерами, когда ее применение позволило получить положительный эффект. После освоения ОТ студентом система переходит к контролю ее применения. В этом состоянии система периодически дает студенту самостоятельно выбирать вариант адаптации учебного элемента для контроля того, что студент все еще сознательно воспроизводит ранее изученную ОТ.

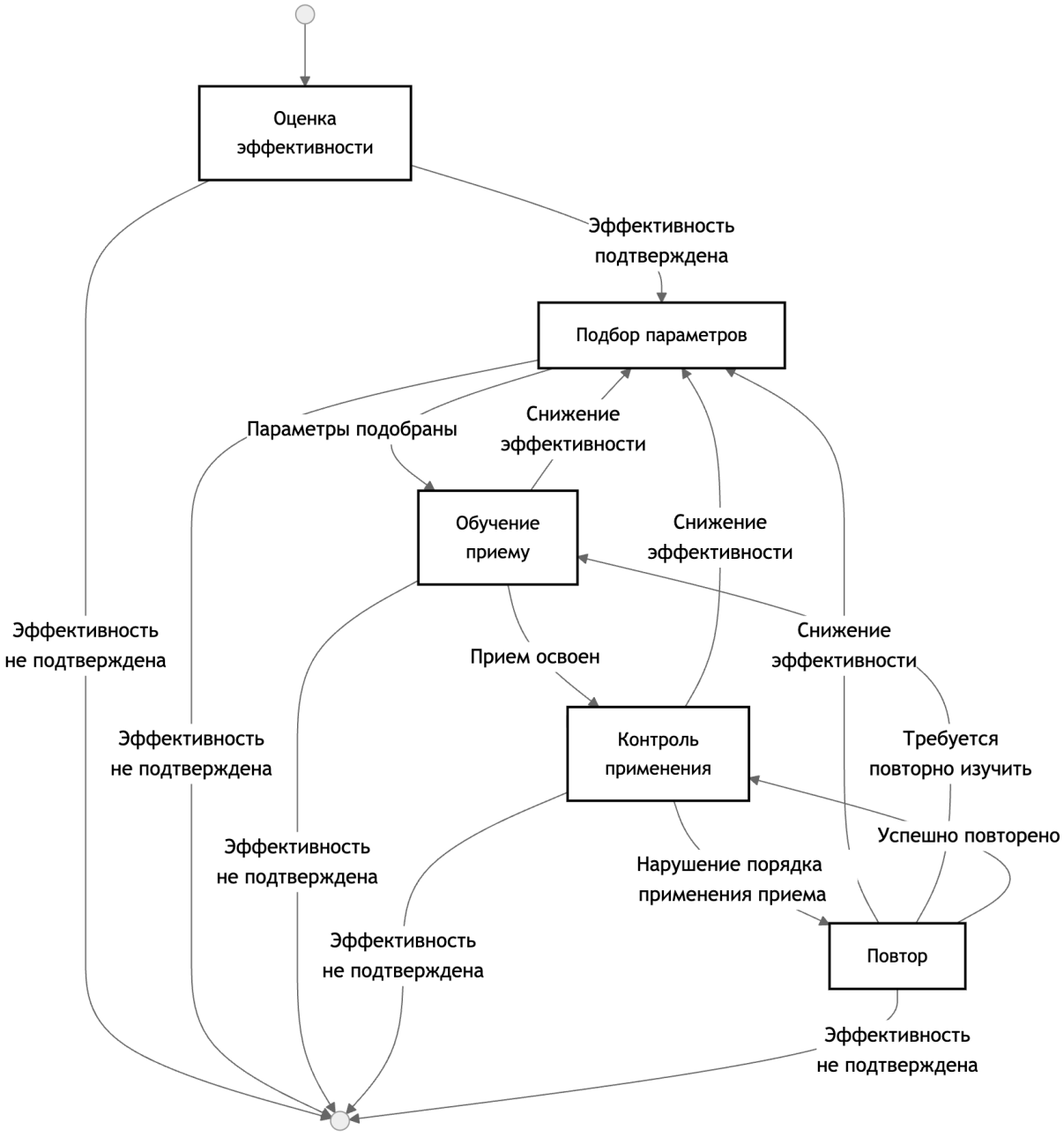


Рис. 1. Жизненный цикл приема обучения.

Материалы изучаемого курса состоят из учебных элементов (УЭ), которые проходят определенные этапы жизненного цикла при взаимодействии со студентом. Студент сначала изучает учебный элемент, после этого происходит закрепление материала и в завершении выполняется повтор материала. Поскольку этапы жизненного цикла учебных элементов могут повторяться из-за необходимости возврата от повтора к закреплению и от закрепления к изучению, жизненный цикл может быть описан в виде автомата с состояниями и условиями перехода между ними (рис. 2).

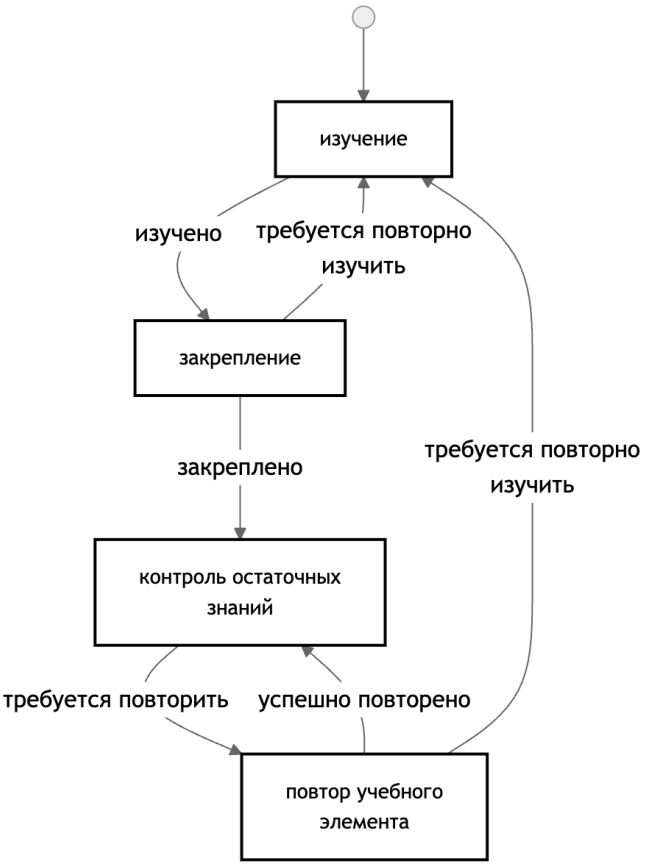


Рис. 2. Жизненный цикл учебного элемента.

Жизненные циклы образовательной технологии и изучаемых учебных элементов связаны. Так, изменение состояний жизненного цикла образовательной технологии происходит на основании данных об изучении учебных элементов, причем образовательная технология сразу охватывает все учебные элементы и при анализе эффективности использует полную информацию, не ограничиваясь контролем усвоения одного учебного элемента.

Другим источником данных для жизненного цикла образовательной технологии является поведение студента. Система контролирует действия студента, стремится обучить его самостоятельно применять образовательную технологию, а после этого выполняет периодический контроль корректности использования ОТ.

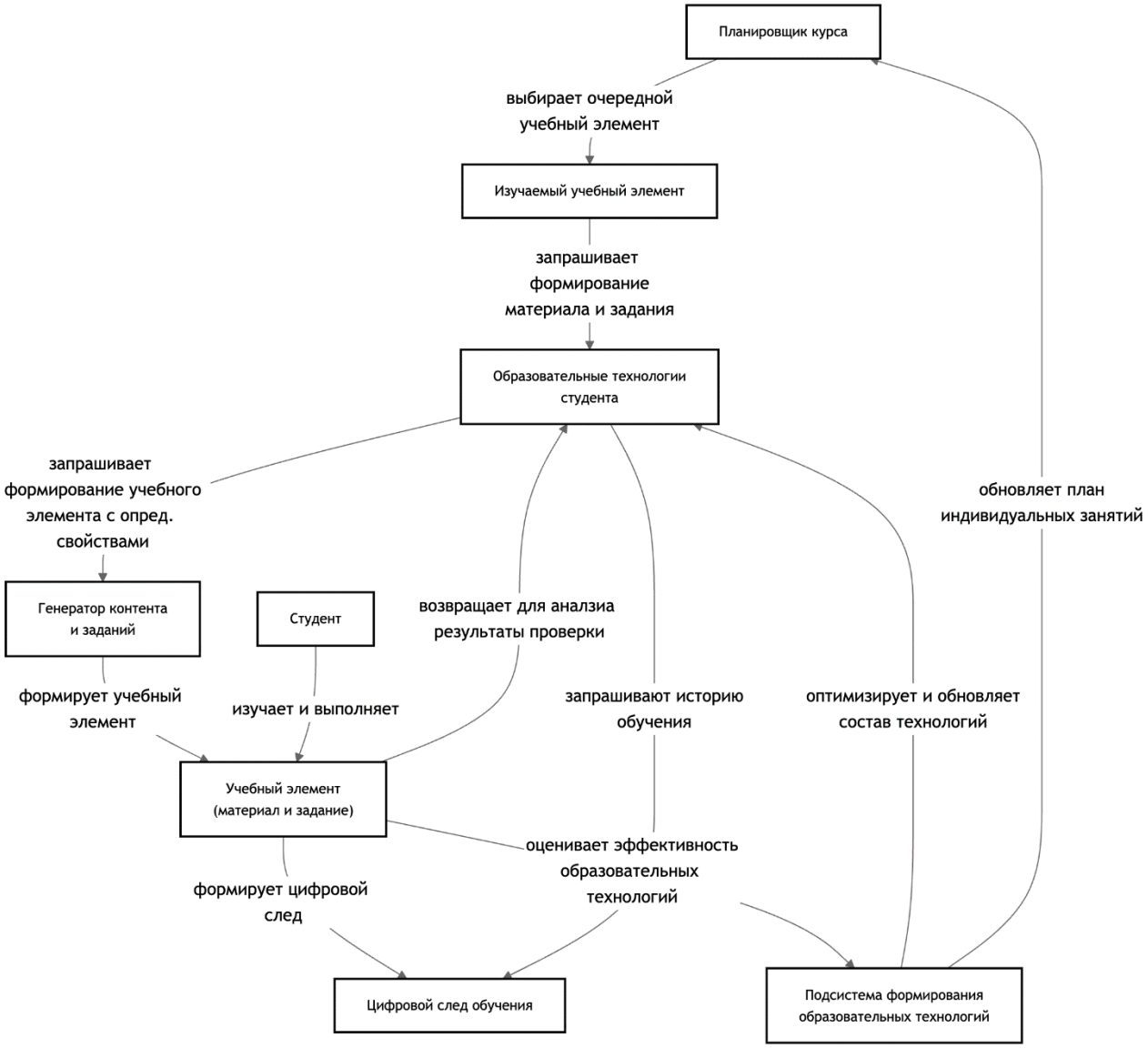


Рис. 3. Диаграмма взаимодействия компонентов системы при формировании ИОТ.

Индивидуальная образовательная технология формируется как результат синтеза на основе подмножества образовательных технологий, отобранных для студента, истории его обучения в виде цифрового следа обучения и адаптивных учебных материалов (рис. 3). Образовательные технологии формируют учебный

элемент, который предъявляется студенту для изучения, после чего выполняется контроль полученных знаний.

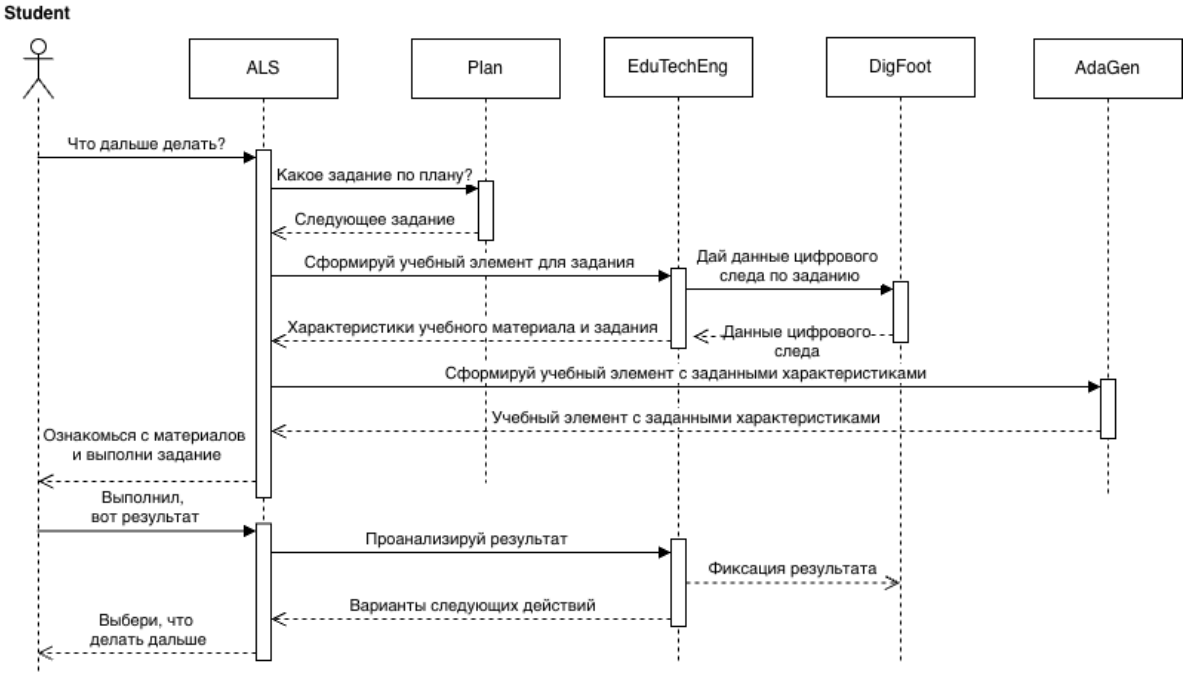


Рис. 4. Диаграмма последовательности при формировании ИОТ. Student – студент, ALS – автоматизированная система обучения, Plan – планировщик, EduTechEng – подсистема обработки ОТ, DigFoot – регистратор цифрового следа студента, AdaGen – подсистема формирования учебного элемента.

Диаграмма последовательности на этапе формирования ИОТ для студента представлена на рис. 4. Инициатором взаимодействия с автоматизированной системой обучения является студент, который обращается к системе после получения от нее уведомления или согласно индивидуальному графику расписания учебы. Система (ALS на рис. 4) получает от планировщика (Plan на рис. 4) информацию о следующем учебном элементе. Данные об учебном элементе передаются в подсистему образовательных технологий, по результатам применения которых формируется набор характеристик для представления учебного элемента. Полученные характеристики передаются в подсистему адаптации, результаты работы которой предоставляются студенту. Студент может ознакомиться с материалами и выполнить задания. По результатам анализа ответов на задание определяется следующее действие. В зависимости от стадии освоения ОТ учебное действие формируется автоматически, либо студенту предлагают выбрать

возможные варианты следующих учебных действий. В зависимости от результата шаги повторяются для текущего или следующего учебных элементов.

Рассмотрим пример ОТ на этапе изучения учебного элемента, основанной на адаптации сложности проверочного задания и изменении уровня детализации разбора задания. Оба этих подхода могут применяться в рамках комплекса ОТ студента, как последовательно, так и в виде единой ОТ.

В качества примера приведем далее описание образовательной технологии этапа изучения УЭ в виде набора правил

Правило 1. Задание выполнено без ошибки, признаков заимствований нет.

Действие 1.1. Запланировать повтор для закрепления материала.

Правило 2. Задание выполнено с ошибкой (первый раз после изучения или успешного выполнения задания), не исключаем случайную ошибку.

Действие 2.1. Ошибка в материалах последнего занятия – запланировать повтор материала в кратком изложении.

Действие 2.2. Ошибка в материале предыдущих занятий – запланировать повтор материала в кратком изложении.

Правило 3. Повторная ошибка в новом материале – вероятно, не понял материал или задачу.

Действие 3.1. Допустим, не понял материал – дать возможность детально посмотреть еще раз материал.

Действие 3.2. После ознакомления с материалом предлагается вопрос из теории.

Вариант действия 3.2.1. Прошел теорию – предлагаем решить задачу.

Вариант действия 3.2.2. Не прошел теорию – повышаем детализацию теории для повторения.

Вариант действия 3.2.3. Не прошел теорию снова – отправляем уведомление преподавателю.

Правило 4. Повторная ошибка после повторного ознакомления с полным материалом – скорее всего не понял задачу.

Действие 4.1. Надо разобрать задачу.

Правило 5. Снова ошибся – интерактивный пошаговый разбор задачи

с контролем ответа на каждом шаге задачи. В итоге нужна диагностика, в чем именно заключается ошибка.

Возможности по модификации и контролю учебных материалов ограничены уровнем их адаптивности. Поскольку подход допускает поэтапную трансформацию материалов курса, часть материалов на момент их использования может не поддерживать необходимую модификацию. В этом случае также не может быть выполнена оценка эффективности применения ОТ. Данный сценарий характерен для старших курсов вузов с малой численностью студентов в группе, для которых нецелесообразно разрабатывать адаптивные учебные материалы, поэтому на этом этапе можно выполнять контроль знаний, но не выбор и подбор значений параметров образовательных технологий. В свою очередь, для младших курсов вуза с большой численностью потока целесообразно разрабатывать адаптивные учебные материалы, поэтому на этих курсах можно применять оценку эффективности и подбор значений параметров образовательных технологий.

Так, например, диаграмма, приведенная на рис. 2, показывает, что контроль и повтор изучения могут происходить в комбинированных заданиях, что снижает нагрузку на студента, однако без возможности детального анализа результатов такая комбинация приведет к тому, что все ОТ, входящие в комбинированную задачу, в случае ошибки потребуется повторять.

Например, для снижения нагрузки студенту могут предложить выполнить повтор нескольких учебных элементов в одном задании, однако в случае ошибки в этом задании система должна или выявить точное место ошибки, или запланировать отдельную проверку знаний по всем учебным элементам, входящим в задание.

Предложенная реализация является расширением цифровой образовательной платформы (ЦОП) Мирера[7]. Формирование цифрового следа обучения, адаптация учебных элементов, их визуализация и проверка знаний выполняются на стороне ЦОП Мирера. Описание образовательных технологий для студента и их применение относятся к зоне ответственности разработанного расширения.

[7]Стартовая страница ЦОП «Мирера». https://www.mirera.ru/

За счет использования готовой функциональности ЦОП Мирера удалось сократить объем работ по внедрению предложенного подхода в учебный процесс.

РЕАЛИЗАЦИЯ

Основу образовательной технологии составляет формально описанный педагогический эксперимент. В качестве описания наиболее наглядной формой является отображение в виде конечного автомата с набором состояний и условий перехода. Альтернативным вариантом является описание в виде скрипта на алгоритмическом языке. В качестве базового языка выбран простой для освоения и распространенный язык разработки Python. Для снижения порога входа преподаватель фиксирует только действия и условия переходов, но не реализует машины состояний.

В скрипте разработчику доступны интерфейсы для получения текущего состояния, включая всю историю обучения и данные группы, интерфейсы для формирования учебного элемента и анализа результатов проверки знаний (листинг 2).

Листинг 1: Фрагмент кода реализации элемента образовательной технологии на языке Python

```
1. if topics.current.attempts.tail(5).succeeded_percentage(80):
2.   # посмотреть сложность последних 5 попыток, должна быть выше целевой
3.  if topics.current.attempts.tail(5).difficulty_percentage(80) >= topics.current.group.target:
4.    # необходимый уровень сложности достигнут, перейти к закреплению
5.    if topics.current.tail_between_days(0, 30).uniorm_distribution_succeeded_percentage(80) >= 80:
6.      pass
7.    else:
8.      topics.current.sustain_in_days(3)
9.    return
10.  else:
11.    # повышаем сложность
12.    topics.current.increase_difficulty()
13.    return
14. elif topics.current.attempts.tail(3).failed_percentage(80):
15.  # может быть случайной ошибкой
16.  topics.current.decrease_difficulty()
```

Поскольку ОТ описана в терминах попыток, программный интерфейс поддерживает методы статистического описания результатов процесса обучения, что делает код ОТ наглядным и снижает риски некорректной реализации типовых функций.

Отдельно описаны критерии оценки применимости и оценки уровня освоения образовательной технологии (листинг 2). Для таких оценок используются попытки, распределение которых важно для принятия решения по эффективности ОТ.

Листинг 2: Фрагмент кода реализации критериев применимости на языке Python

```
1. def could_be_effective(self, topics):
2.   return topics.all.attempts.last(10).
         right_distribution_failed_percentage(80)
3. def is_effective(self, topics):
4.   return topics.current.attempts.last(10).
         right_distribution_failed_percentage(80)
```

Разработчик ОТ также может принять во внимание особенности работы в группе, зафиксировав правила формирования ИОТ с учетом динамики всей группы. Учет групповой динамики важен, поскольку работа происходит в рамках учебной программы курса. Например, если у всей группы возникает затруднение с каким-либо материалом, то его надо повторять сразу со всеми студентами, а не отдельно с каждым.

При формировании ИОТ соблюдаются следующие принципы.

1. Достижимость целей программы обучения: с учетом предложенного набора приемов и их параметров студент должен выполнять учебную программу согласно расписанию группы, чтобы завершить обучение в срок вместе с группой. Вариант досрочного завершения обучения в контексте обучения в группе не рассматривается, поскольку очные занятия синхронны для всей группы.

2. Минимизация трудозатрат студента: количество часов, затрачиваемых на подготовку, не должно превышать нагрузку, заложенную в программу. Однако студент может тратить меньше времени на подготовку к занятиям.

3. Минимальное количество и сложность образовательных технологий, используемых для построения ИОТ: при формировании индивидуального набора образовательных технологий необходимо стремиться к созданию минимального набора простых методов, подходящих для самостоятельного использования.

4. Универсальный набор образовательных технологий для построения ИОТ всех дисциплин: отличительной особенностью подхода является то, что набор образовательных технологий подбирается сразу для всех дисциплин, изучаемых студентом, при этом допускается, что не все образовательные технологии используются в каждой дисциплине.

5. Отсутствие фазы накопления данных и возможность холодного старта: ИОТ начинает формироваться сразу после начала обучения, начальный набор требуемых образовательных технологий может быть сформирован преподавателем при подготовке курса.

Одним из вариантов ускорения поиска ИОТ с учетом итерационного характера ее поиска является укрупнение ОТ. Для успешного применения образовательных технологий в системе необходимо обеспечить соответствие между сложностью, выраженной в количестве контролируемых аспектов образовательной технологии (приема обучения), и возможностями по контролю эффективности применения отдельных аспектов ОТ. Объединение нескольких простых ОТ в одну составную ОТ целесообразно, если в системе есть возможность оценить влияние на результат каждой ОТ в отдельности. При этом важно учитывать, что методы модификации учебных материалов и контроля результатов, комбинируемых ОТ, не должны конфликтовать (предлагать различные варианты адаптации учебного элемента). В базовой стратегии применения одна ОТ соответствует целому этапу, такому как изучение, закрепление и повтор (см. рис. 2). В более сложной стратегии ОТ могут комбинироваться по непротиворечивости формирующих воздействий и контрольных метрик. ОТ также может являться результатом направленного поиска индивидуальной комплексной ОТ для студента, которая будет представлять собой результат автоматического синтеза нового приема из нескольких существующих.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Описанный подход отличается от отслеживания или трассировки знаний, поскольку не предполагает оценки состояния знаний студента, хотя не исключает возможность оценки знаний по данным цифрового следа обучаемого, который при частой проверке знаний содержит актуальный срез знаний обучаемого. Применение модели памяти обучаемого на этапе повторения учебного элемента дает возможность также оценить, какие знания находятся в активном состоянии.

Предложенная система изначально предназначена для развития навыков самообразования. В процессе использования системы студент помимо освоения учебных материалов также обучается эффективным приемам самообучения. Рассмотренные выше исследования проводились для оценки эффективности алгоритмов обучения, но исследователи не оценивали развитие навыков самообучения при взаимодействии студентов с системой. В результате студент выполняет действия заложенного алгоритма, но не осознает шаги и особенности применяемой ОТ. В исследовании [20] студент самостоятельно определяет стратегию обучения для достижения своих целей, однако в предлагаемом подходе автоматизированная система помогает найти индивидуальную стратегию, обучает студента ее применению, что дает возможность контролировать не только результат обучения, но и способ получения результата.

Результаты обучения зависят от уровня мотивации студента и могут быть искажены в случае случайных ошибок и заимствований готовых решений. Проблема случайных ошибок решается на уровне образовательной технологии, которая должна быть устойчива к таким событиям. Проблема заимствования готовых ответов может быть решена генерацией уникальных заданий, а также проверкой на заимствование ответов по аналогичным заданиям в группе. В исследовании [20] предложено создать конкурентную среду внутри группы, в результате стратегия «поделиться результатом» для студента, ответившего правильно, становится невыгодной, что снижает мотивацию для разглашения ответов заданий.

В работе [22] рассмотрена связь адаптивного обучения с универсальными учебными (регулятивными) действиями. В отличие от универсального учебного действия предлагаемые автором образовательные технологии изначально

представляются комплексными понятиями, которые ближе по смыслу к приемам обучения как к части метода обучения, чем к атомарным универсальным учебным действиям. Однако разработчики образовательных технологий могут в качестве образовательных технологий реализовывать приемы обучения, направленные на освоение универсальных учебных действий. Противоречий здесь нет.

Отделение образовательных технологий от материалов курса с возможностью переносить ОТ на следующий или параллельный курс приближает предложенный подход к изначальной архитектуре адаптивного обучения, состоящей из моделей обучаемого, предметной области и самого обучения. В предлагаемом нами подходе модель предметной области представляет собой непосредственно адаптивные учебные материалы, модель студента – цифровой след обучения и подмножество образовательных технологий, а модель обучения – все множество образовательных технологий. Таким образом, в системе происходят сразу две адаптации: 1) на уровне учебных материалов, 2) адаптации образовательных технологий к особенностям и потребностям студента.

ЗАКЛЮЧЕНИЕ

Предложенные подход и его реализация обладают хорошей интерпретируемостью, поскольку основаны на результатах хорошо интерпретируемых педагогических экспериментов. Преподаватели могут быть вовлечены в разработку и доработку готовых образовательных технологий, т.к. система предоставляет программный интерфейс для встраивания своих решений без привлечения разработчиков. Опыт Plario показывает, что можно сформировать группу преподавателей для разработки учебных курсов, в данном случае – образовательных технологий.

В предложенном подходе формируемое подмножество ОТ ассоциировано со студентом и может переноситься на следующий и параллельный курсы, а также использоваться после вуза на курсах повышения квалификации. Такое отделение ОТ также дает возможность оценивать применимость образовательных технологий на всех автоматизированных курсах, на которых обучается студент, ускоряя процесс поиска индивидуального подмножества ОТ.

Описание модели обучения в виде набора образовательных технологий также дает возможность оценивать сроки завершения обучения с учетом индивидуальных параметров, подобранных алгоритмом, что позволяет на раннем этапе выявить студентов, которые потенциально не смогут успешно завершить курс.

Построенная система по сути является для студента рекомендательной – она помогает студенту осваивать материал конкретного курса и развивать навыки самообучения. Полученные навыки самообучения используются студентом там, где недоступны адаптивные технологии, например на специализированных курсах, для которых создание адаптивных курсов сейчас нецелесообразно из-за незначительной численности обучаемых.

Описание автоматизированных учебных технологий остается сложной задачей, доступной только преподавателям-исследователям. Однако масштабирование ее использования может быть реализовано с низким порогом входа без необходимости выполнять сложные конфигурирование или разрабатывать программные модули для автоматизированной системы. Описание образовательных технологий на языке Python дает низкий порог вхождения за счет простоты интерпретации описанного алгоритма.

Для реализации подобных технологий требуется большое количество проработанных учебных материалов различной степени детализации, с пошаговыми разборами и примерами. Важно также обеспечить вариативность заданий, чтобы снизить вероятность заимствования. Такой объем работы может быть выполнен только группой специалистов при подготовке типового курса.

Качественные учебные материалы все еще остаются ключевым фактором успеха при внедрении технологий индивидуализированного обучения. В предлагаемом подходе учебные материалы состоят из учебных элементов, которые разрабатываются централизованно группой преподавателей или экспертов, как в системе Plario, что снижает индивидуальные трудозатраты каждого преподавателя при создании курса. Каждый учебный элемент должен поддерживать адаптивность, а также вариативность проверочных заданий. Основные трудозатраты сосредоточены на начальном этапе разработки учебного курса. Адаптация курса под программу конкретного учебного заведения является менее трудоем-

кой задачей за счет использования готовых блоков. Преимущество предлагаемого подхода состоит в том, что при его использовании не требуется сразу перерабатывать весь курс, а допускается постепенно трансформировать курс, начав с тем, вызывающих у студентов затруднения при изучении.

Для специализированных предметов старших курсов разработка учебных материалов в силу их узкой специализации является обязанностью отдельных преподавателей, но в этом случае снижаются требования к адаптивности и вариативности этих материалов, так как студенты должны активно использовать приемы самостоятельного обучения, освоенные на младших курсах.

Поскольку образовательные технологии отделены от учебных материалов, разработанные курсы ограниченно переносимы. Эти ограничения рассмотрены нами в работе [21].

Несмотря на активное развитие технологий генеративного искусственного интеллекта в образовании, вопросы качества учебных материалов находятся в зоне ответственности преподавателей, поэтому все результаты, приведенные выше, применимы также к использованию генеративного искусственного интеллекта.

Предложенная реализация накладывает определенные требования к учебному курсу не только с точки зрения гибкости учебных материалов, но и с позиции структуры проверочных заданий. Для снижения трудозатрат обучаемого на прохождение проверочных мероприятий задания должны становиться комплексными, чтобы студент показывал не только понимание только что завершенной темы, но и подтверждал, что он все еще понимает материал, изученный ранее (закрепление). Если подобные задания не получится внедрить в курс без существенного изменения его структуры, то такие задания должны выполняться студентом в рамках самостоятельной работы, например, при повторении ранее пройденного материала.

Благодарности

Работа выполнена в рамках темы государственного задания НИЦ «Курчатовский институт» – НИИСИ по теме № FNEF-2024-0001 (1023032100070-3-1.2.1).

СПИСОК ЛИТЕРАТУРЫ

1. *Hoda H., Sujo-Montes L., Tu C.-H., Armfield S.J.W., and Yen C.-J.* Assessment and Learning in Knowledge Spaces (ALEKS) Adaptive System Impact on Students' Perception and Self-Regulated Learning Skills // Education Sciences. 2021. Vol. 11. No. 10. Article 603. https://doi.org/10.3390/educsci11100603

2. *Lopes R.P., Mesquita C., de Góis L.A., dos Santos G. Júnior.* Students' learning autonomy: a systematic literature review // EDULEARN19 Proceedings. 2019. P. 5958–5964. https://doi.org/10.21125/edulearn.2019.1435

3. *Овезова У.А., Вагнер М.-Н.Л.* Формирование навыков самообразовательной деятельности студентов в условиях дистанционного образования // Мир науки, культуры, образования. 2021. -№ 2(87). С. 160–162. https://doi.org/10.24412/1991-5497-2021-287-160-162

4. *Вайнштейн Ю.В., Есин Р.В., Цибульский Г.М.* Модель образовательного контента: от структурирования понятий к адаптивному обучению // Открытое образование. 2021. Т. 25. № 1. С. 28–39. EDN: CODQHI. https://doi.org/10.21686/1818-4243-2021-1-4-28-39

5. *Комлева Н.В., Вилявин Д.А.* Цифровая платформа для создания персонализированных адаптивных онлайн курсов // Открытое образование. 2020. Т. 24. № 2. С. 65–72. EDN: CWBDOO. https://doi.org/10.21686/1818-4243-2020-2-65-72

6. *Шамсутдинова Т.М.* Формирование индивидуальной образовательной траектории в адаптивных системах управления обучением // Открытое образование. 2021. Т. 25. № 6. С. 36-44. EDN: YPLVRY. https://doi.org/10.21686/1818-4243-2021-6-36-44

7. *Павлов А.Ф., Мякишева Ю.В., Родионова Г.Н.* Опыт применения технологии адаптивного обучения в образовательном процессе высшего учебного заведения // Известия Самарского научного центра Российской академии наук. Социальные, гуманитарные, медико-биологические науки. 2024. Т. 26. № 3(96). С. 70–76. EDN: HFAUJI. https://doi.org/10.37313/2413-9645-2024-26-96-70-76

8. *Подколзин М.М.* Интеллектуальная система адаптивного обучения на основе нейронных сетей для персонализации образовательных траекторий студентов российских вузов // Информатика и образование. 2024. Т. 39 № 6. С. 65–81. https://doi.org/10.32517/0234-0453-2024-39-6-65-81

9. *Зарипова З.Ф.* PLARIO как цифровой инструмент повышения уровня математической подготовки студентов-бакалавров первого курса // Казанский педагогический журнал. 2023. № 3(158). С. 131–139. EDN: ZDVOKN. https://doi.org/10.51379/KPJ.2023.160.3.017

10. *Zhilmagambetova R. et al.* The Role of Adaptive Personalized Technologies in the Learning Process: Stepik as a Tool for Teaching Mathematics // International Journal of Virtual and Personal Learning Environments (IJVPLE). 2023. Vol. 13. No. 1. P. 1–15. https://doi.org/10.4018/IJVPLE.324079

11. *Садыкова А.Р., Трухманов Д.В.* Адаптивное обучение с использованием нейронных сетей: опыт и перспективы // Вестник МГПУ. Серия «Информатика и информатизация образования». 2025. № 2(72). С. 32–45. https://doi.org/10.24412/2072-9014-2025-272-32-45

12. *Шудуева З.А.,* Миназова З.М., Харченко С.Б. Роль адаптивных образовательных технологий в персонализации обучения // Проблемы современного педагогического образования. 2024. №84-1. С. 379–382. EDN: GISEIN.

13. *Qiu Y, Asniza IN, Zheng S.* The development of mathematics lesson plan integrated with Squirrel artificial intelligence in enhancing the fifth grade primary school students' cognitive development in Fujian Province, China // InProceedings of the 2024 3rd International Conference on Artificial Intelligence and Education. 2024. P. 308–313. https://doi.org/10.1145/3722237.3722290

14. *Вилкова К.А., Лебедев Д.В.* Адаптивное обучение в высшем образовании: за и против // Национальный исследовательский университет «Высшая школа экономики». Институт образования. М.: НИУ ВШЭ, 2020. 36 с. EDN: PYRWTW.

15. *Heift T.* Web Delivery of Adaptive and Interactive Language Tutoring: Revisited // International Journal of Artificial Intelligence in Education. 2015. Vol. 26. P. 489–503. https://doi.org/10.1007/s40593-015-0061-0

16. *Кречетов И.А., Романенко В.В.* Реализация методов адаптивного обучения // Вопросы образования. 2020. № 2. С. 252–277. EDN: KYNIIH. https://doi.org/10.17323/1814-9545-2020-2-252-277

17. *Murre J.M.J., Dros J.* Replication and Analysis of Ebbinghaus' Forgetting Curve // PLoS ONE. Vol. 10. No. 7. Article e0120644. https://doi.org/10.1371/journal.pone.0120644

18. *Piech C., Bassen J., Huang J. et al.* Deep Knowledge Tracing // Proc. 28th Int. Conf. Neural Information Processing Systems (NIPS 2015)). 2015. No. 1. P. 505–515.

19. *Abdelrahman G., Wang Q., Pereira N.B.* Knowledge tracing: A survey // ACM Computing Surveys. 2022. Vol. 55. No. 11. Article 224. P. 1–37. https://doi.org/10.1145/3569576

20. *Чупин Н.А.* Конкурентная система оценивания учебных достижений студентов в контексте адаптивного обучения // Научно-методический электронный журнал «Концепт». 2023. № 6. С. 44–62. https://doi.org/10.24412/2304-120X-2023-11047

21. *Дьяченко М.С., Леонов А.Г.* Решение задачи автоматизации учебного процесса с помощью экспериментального поиска индивидуальной образовательной траектории // Информатика и образование. 2024. Т. 39. № 4. С. 14–26. EDN: BMEVBI. https://doi.org/10.32517/0234-0453-2024-39-4-14-26

22. *Фисенко Т.П.* Организация адаптивного обучения, направленного на развитие регулятивных универсальных учебных действий обучающихся основной школы (на примере обучения математике) // Вестник Омского государственного педагогического университета. Гуманитарные исследования. 2024. № 4(45). С. 215–220. https://doi.org/10.36809/2309-9380-2024-45-215-220

ALGORITHMS FOR INDIVIDUALIZING LEARNING BASED ON THE COMPOSITION OF THE RESULTS OF PEDAGOGICAL EXPERIMENTS

M. S. Diachenko[0000-0002-5809-4981]

National Research Centre “Kurchatov Institute”, Moscow, Russia

mdyachenko@vip.niisi.ru

Abstract

This paper presents various aspects of the practical implementation of individualized learning algorithms (based on the results of pedagogical experiments) for both teacher-led instruction (in the classroom, remotely, or in a hybrid mode) and independent student work. The described system simultaneously teaches students course materials and independent learning techniques — that is, educational technologies

that shape an individualized educational trajectory. A subset of educational technologies is determined individually for each student in the group. The educational technologies are independent of the course and universal, so they can be applied in subsequent or parallel courses. Teachers can describe new educational technologies as Python scripts without the involvement for developers. The proposed implementation integrates with the Mirera digital educational platform to expand the platform's capabilities.

***Keywords**: individualization of learning, automated learning system, digital educational platform, adaptive learning.*

REFERENCES

1. *Hoda H., Sujo-Montes L., Tu C.-H., Armfield S.J.W., and Yen C.-J.* Assessment and Learning in Knowledge Spaces (ALEKS) Adaptive System Impact on Students' Perception and Self-Regulated Learning Skills // Education Sciences. 2021. Vol. 11. No. 10. Article 603. https://doi.org/10.3390/educsci11100603

2. *Lopes R.P., Mesquita C., de Góis L.A., dos Santos G. Júnior.* Students' learning autonomy: a systematic literature review // EDULEARN19 Proceedings. 2019. P. 5958–5964. https://doi.org/10.21125/edulearn.2019.1435

3. *Ovezova U.A., Wagner M.-N.L.* The formation of students' self-educational skills in the context of distance education // The world of science, culture and education. 2021. No. 2(87). P. 160–162 (In Russian).
https://doi.org/10.24412/1991-5497-2021-287-160-162

4. *Vainshtein J.V., Esin R.V., Tsibulsky G.M.* Learning content model: From concept structuring to adaptive learning. Open Education. 2021. Vol. 25. No. 1. P. 28–39. (In Russian) EDN: CODQHI. https://doi.org/10.21686/1818-4243-2021-1-4-28-39

5. *Komleva N.V., Vilyavin D.A.* Digital platform for creating personalized adaptive online courses. Open Education. 2020. Vol. 24, No. 2. P. 65–72 (In Russian). EDN: CWBDOO. https://doi.org/10.21686/1818-4243-2020-2-65-72

6. *Shamsutdinova T.M.* Formation of individual educational trajectory in adaptive learning management system. Open Education. 2021. Vol. 25. No. 6. P. 36–44 (In Russian). EDN: YPLVRY. https://doi.org/10.21686/1818-4243-2021-6-36-44

7. *Pavlov A.F., Myakisheva Yu.V., Rodionova G.N.* The use of adaptive learning technology in the educational process of higher education institutions // Izvestia of

the Samara Science Centre of the Russian academy of science. Social, humanitarian, medicobiological sciences. 2024. Vol. 26. No. 3(96). P. 70–76 (In Russian). EDN: HFAUJI. https://doi.org/10.37313/2413-9645-2024-26-96-70-76

8. *Podkolzin M.M.* Intelligent Adaptive Learning System Based on Neural Networks for Personalization of Educational Trajectories of Russian University Students. Informatics and Education. 2024. Vol. 39 No. 6. P. 65–81 (In Russian). https://doi.org/10.32517/0234-0453-2024-39-6-65-81

9. *Zaripova Z.F.* PLARIO as a digital tool for improving the level of mathematical training of first-year undergraduate students // Kazan Pedagogical Journal. 2023. No. 3(158). P. 131–139 (In Russian). EDN: ZDVOKN. https://doi.org/10.51379/KPJ.2023.160.3.017

10. *Zhilmagambetova R. et al.* The Role of Adaptive Personalized Technologies in the Learning Process: Stepik as a Tool for Teaching Mathematics // International Journal of Virtual and Personal Learning Environments (IJVPLE). 2023. Vol. 13. No. 1. P. 1–15. https://doi.org/10.4018/IJVPLE.324079

11. *Sadykova A.R., Trukhmanov D.V.* Adaptive learning using neural networks: experience and prospects // MCU Journal of Informatics and Informatization of Education. 2025. No. 2(72). P. 32–45 (In Russian). https://doi.org/10.24412/2072-9014-2025-272-32-45

12. *Shudueva Z.A., Minazova Z.M., Kharchenko S.B.* The role of adaptive educational technologies in the personalization of learning // Problems of modern teacher education. 2024. No. 84-1. P. 379–382 (In Russian). EDN: GISEIN.

13. *Qiu Y, Asniza I.N., Zheng S.* The development of mathematics lesson plan integrated with Squirrel artificial intelligence in enhancing the fifth grade primary school students' cognitive development in Fujian Province, China // In Proceedings of the 2024 3rd International Conference on Artificial Intelligence and Education. 2024. P. 308–313. https://doi.org/10.1145/3722237.3722290

14. *Vilkova K.A., Lebedev D.V.* Adaptive learning in higher education: Pro et contra // Moscow, National Research University Higher School of Economics; 2020. 36 p. (In Russian). EDN: PYRWTW.

15. *Heift T.* Web Delivery of Adaptive and Interactive Language Tutoring: Revisited // International Journal of Artificial Intelligence in Education. 2015. Vol. 26. P. 489–503. https://doi.org/10.1007/s40593-015-0061-0

16. *Krechetov I.A., Romanenko V.V.* Implementing the adaptive learning techniques // Educational Studies. Moscow. 2020. No. 2. P. 252–277 (In Russian). EDN: KYNIIH. https://doi.org/10.17323/1814-9545-2020-2-252-277

17. *Murre J.M.J., Dros J.* Replication and Analysis of Ebbinghaus' Forgetting Curve // PLoS ONE. Vol. 10. No. 7. Article e0120644.
https://doi.org/10.1371/journal.pone.0120644

18. *Piech C., Bassen J., Huang J. et al.* Deep Knowledge Tracing // Proc. 28th Int. Conf. Neural Information Processing Systems (NIPS 2015)). 2015. No. 1. P. 505–515.

19. *Abdelrahman G., Wang Q., Pereira N.B.* Knowledge tracing: A survey // ACM Computing Surveys. 2022. Vol. 55. No. 11. Article 224. P. 1–37.
https://doi.org/10.1145/3569576

20. *Chupin N.A.* Competitive assessment system of students' academic achievements in the context of adaptive learning // Scientific-methodological electronic journal "Koncept". 2023. No. 6. P. 44–62 (In Russian).
https://doi.org/10.24412/2304-120X-2023-11047

21. *Diachenko M.S., Leonov A.G.* Solving the problem of automating the learning process through experimental search for an individual educational trajectory // Informatics and education. 2024. Vol. 39. No. 4. P. 14–26 (In Russian). EDN: BMEVBI. https://doi.org/10.32517/0234-0453-2024-39-4-14-26

22. *Fisenko T.P.* Organization of Adaptive Learning Aimed at the Development of Regulatory Universal Educational Actions of Primary School Students (The Case of Teaching Mathematics) // Bulletin of the Omsk State Pedagogical University. Humanitarian studies. 2024. No. 4(45). P. 215–220 (In Russian).
https://doi.org/10.36809/2309-9380-2024-45-215-220

СВЕДЕНИЯ ОБ АВТОРЕ

ДЬЯЧЕНКО Михаил Сергеевич – закончил магистратуру в Московском государственном техническом университете «СТАНКИН» в 2006 г., в 2023 окончил аспирантуру по направлению 09.06.01 Информатика и вычислительная техника в НИИСИ РАН. В настоящее время инженер ОПИБ НИИСИ РАН, г. Москва. Область научных интересов: системы адаптивного обучения, распределенные системы управления.

Mikhail Sergeevich DIACHENKO – completed master's degree at the Moscow State Technical University "STANKIN" in 2006 and doctoral program in 2023 at the Scientific Research Institute of System Analysis and Control (SRISA RAS), specializing in Computer Science and Engineering (09.06.01). Currently an engineer at SRISA RAS, Moscow. Research interests: adaptive learning systems and distributed control systems.

email: mdyachenko@niisi.ru
ORCID: 0000-0002-5809-4981

Материал поступил в редакцию 4 марта 2026 года

УДК 013+004.65

АДМИНИСТРИРОВАНИЕ КОНТЕНТА ЭЛЕКТРОННОЙ БИБЛИОТЕКИ «НАУЧНОЕ НАСЛЕДИЕ РОССИИ»

Н. Е. Каленов[1] [0000-0001-5269-0988], К. П. Погорелко[2] [0000-0002-4778-3060]

[1, 2]*Национальный исследовательский центр «Курчатовский институт», г. Москва, Россия*

[1]nekalenov@mail.ru, [2]konstpog@yandex.ru

Аннотация

Электронная библиотека «Научное наследие России» (ЭБ ННР) функционирует в открытом доступе в Интернете начиная с 2010 г. Библиотека интегрирует информацию об ученых, внесших вклад в развитие российской науки, их научных публикациях, связанных с ними архивных материалах, сетевых ресурсах и музейных предметах. Современная версия ЭБ ННР развивается как модель фрагмента Единого цифрового пространства научных знаний (ЕЦПНЗ) и включает ряд функциональных блоков (формирование метаданных, публикация оцифрованных документов и музейных предметов, организация коллекций и выставок, администрирование контента).

В статье описана функциональность административного блока электронной библиотеки. Работа с блоком доступна авторизованным пользователям, имеющим соответствующие права. Блок обеспечивает возможность редактирования элементов метаданных объектов каждого типа и связей между ними, мониторинг этапов обработки конкретных объектов, вводимых в ЭБ, позволяет экспортировать заданный набор связанных данных в формате *.csv. Представлены экранные формы блока и примеры работы с ним.

***Ключевые слова**: цифровая библиотека, научное наследие, управление, интерфейс поиска, связанные данные.*

ВВЕДЕНИЕ

С 2010 г. электронная библиотека «Научное наследие России» (далее ЭБ ННР) находится в свободном доступе в Интернете. Отличительной чертой библиотеки, что делает ее уникальной, является объединение связанной между

собой разноплановой информации. Библиотека объединяет информацию об ученых, внесших существенный вклад в развитие российской науки, их основных публикациях (включая оцифрованные тексты), а также связанных с ними архивных материалах и музейных предметах. Временной охват представленной информации включает период, начиная с XVIII в. – первая треть XX в. Подробное описание принципов формирования и наполнения ЭБ ННР представлено в [1].

ЭБ ННР является востребованным ресурсом. Анализ использования библиотеки, выполненный в [2], показал неуклонный рост ее посещаемости и запрашиваемости содержащейся в ней информации, что подтверждает правильность принципов и решений по администрированию контента, положенных в основу проекта.

Работы по созданию и поддержке ЭБ ННР выполнялись в рамках целевой академической программы и финансировались Президиумом РАН через Межведомственный суперкомпьютерный центр (МСЦ) РАН, сотрудники которого обеспечивали ввод, редактирование данных и техническую поддержку функционирования Библиотеки. В формировании контента ЭБ ННР участвовало несколько десятков академических организаций. В 2015 г., в связи с реорганизацией РАН, целевая программа была закрыта, и ввиду отсутствия целевого финансирования большинство участников приостановили участие в проекте. До 2020 г. ЭБ ННР продолжала функционировать за счет грантов РФФИ. В связи с ликвидацией РФФИ «внешняя» финансовая поддержка Библиотеки была полностью прекращена. К этому времени контент ЭБ ННР содержал информацию о нескольких тысячах ученых и включал около 30 тысяч уникальных публикаций. Чтобы не потерять этот ресурс, в 2020 г. руководством МСЦ РАН было принято решение развивать его в соответствии с современными требованиями в качестве модели фрагмента Единого цифрового пространства научных знаний (ЕЦПНЗ) [3–5], исследования в области формирования которого ведутся в МСЦ РАН[1].

[1] В настоящее время МСЦ РАН преобразован в Отделение суперкомпьютерных систем и параллельных вычислений НИЦ «Курчатовский институт», и исследования в области создания ЕЦПНЗ выполняются в рамках государственного задания.

ПРОГРАММНЫЙ КОМПЛЕКС ЭБ НИР

Программный комплекс ЭБ НИР первоначально был разработан на базе платформы Microsoft ASP.NET CORE 3.1 в архитектуре MVC (Model-View-Controller) на языке C# в операционной среде Windows [6]. Эта платформа на момент создания была одной из самых современных и обеспечивала независимость от операционной среды, что оказалось немаловажным фактором при переносе системы в операционную среду Unix [7]. Платформы разработки со временем стареют, и их разработчики перестают выпускать обновления. Поэтому своевременный перевод программного комплекса на новую версию платформы является важным этапом в его развитии и поддержании надежности функционирования на высоком уровне. Соответственно, программный комплекс ЭБ НИР был переведен на .NET 6.0, а затем на платформу .NET 8.0. Перевод программного комплекса на новую платформу осуществлялся путем его полной пересборки с учетом новой платформы и замены всех используемых библиотек на версии, соответствующие версии .NET. Это включало в себя не только обновление библиотек, но и внесение изменений в программный код, чтобы использовать новые функциональные возможности, которые появились в обновленных версиях. В первую очередь, это касалось библиотек, обеспечивающих доступ к данным, что позволило значительно оптимизировать работу с базами данных.

Одними из ключевых изменений стали использование асинхронного режима выполнения запросов и внедрение отложенного доступа к данным. Эти подходы значительно повысили производительность системы и уменьшили нагрузку на сервер. Кроме того, в новой версии платформы в языке C# были введены более строгие правила проверки кода, что потребовало дополнительных усилий для формального уточнения текстов программ и исключения возможных предупреждений о потенциальных ошибках.

В настоящее время функционирование комплекса осуществляется на сервере под управлением операционной системы Ubuntu 22.04.5 с использованием пакета Microsoft.NETCore.App 8.0.21.

РЕШЕНИЯ ПО ОРГАНИЗАЦИИ АДМИНИСТРИРОВАНИЯ ДАННЫХ

Качество и надежность информационной системы во многом зависят от организации технологических процессов по вводу и контролю вводимой информации и, соответственно, программных средств, обеспечивающих эти процессы. Хотя технологические процедуры пополнения, верификации, сохранения и обеспечения доступа сильно отличаются от системы к системе, существуют общие требования, которые, по возможности, должны выполняться. Из современных работ, в которых рассматриваются общие принципы функционирования информационных систем и, в частности, аспекты организации администрирования данных, отметим [8] и [9]. В соответствии с этими подходами и на основе накопленного опыта эксплуатации информационных систем были реализованы представленные ниже решения по администрированию данных ЭБ ННР.

Основной особенностью ЭБ ННР, так же, как и ЕЦПНЗ в целом, является распределенное формирование контента – ввод данных осуществляется широким кругом исполнителей [10]. Поэтому для обеспечения контроля качества вводимых данных был предпринят ряд специальных мер. Применительно к ЭБ ННР данные в процессе ввода проходят несколько стадий. Так, например, такой объект, как публикация, может находиться в одном из следующих состояний:

- предложено к сканированию;
- отклонено;
- зарегистрировано;
- в работе;
- завершено;
- требует исправления;
- опубликовано;
- отказ от сканирования.

Для контроля прав пользователей введен механизм ролей и установлена строгая регламентация того, кто имеет право перевода объектов из одного статуса в другой. В технологии формирования контента ЭБ ННР реализованы следующие принципы.

- Персонифицированность вводимой информации. В системе ведутся протоколы изменений, содержащие данные о том, кто и когда вводил или исправлял данную информацию. Эта мера повышает ответственность исполнителей.
- Для осуществления выходного контроля данных выделен узкий круг ответственных выпускающих редакторов, которые обеспечены достаточным и удобным инструментарием для выявления возможных ошибок.
- Организация и обработка обратной связи с пользователями. Анализ замечаний пользователей входит в обязанности одного из сотрудников, сопровождающих систему.

ФУНКЦИОНАЛЬНОСТЬ АДМИНИСТРАТИВНОГО БЛОКА

Административный блок ЭД ННР обеспечивает:

- возможность просмотра и редактирования контента Библиотеки;
- регистрацию неисправимых силами сотрудников ошибок (отсканированные публикации формируются децентрализовано и могут содержать пропуски страниц или брак, который может быть устранен только при сопоставлении с оригиналом);
- возможность «опубликования» материалов (установки признака, по которому объект становится доступным при поиске в пользовательском интерфейсе);
- возможность исключения объекта из опубликованного массива;
- ввод новых атрибутов объектов и именованных связей;
- ввод внешних ссылок на ресурсы, связанные с отраженными в ЭБ ННР, объектами;
- экспорт выбранных объектов с заданием списка выводимых атрибутов;
- возможность ввода и редактирования раздела «Новости», представленного на начальной странице ЭБ ННР (рис. 1).

Рис. 1. Начальная страница ЭБ ННР.

Для реализации этих функций в блоке имеется мощный поисковый аппарат [11], позволяющий выделять объекты не только по значениям атрибутов и связей, но и по количеству объектов, связанных между собой. В частности, с его помощью можно получить список персон с заданным количеством связанных изображений. Данный поисковый блок разрабатывался с учетом его дальнейшего использования в развивающемся проекте ЕЦПНЗ. Поэтому его привязка к конкретной структуре базы данных, составу объектов, их атрибутов, а также к возможным связям между объектами и атрибутами связей осуществляется путем задания соответствующих параметров и не требует настройки программного кода.

Доступ к административному блоку имеют авторизованные пользователи, имеющие разные права. Максимальными правами обладают пользователи со статусом «администратор», которым доступны все операции, включая регистрацию новых пользователей и установление их прав.

На рис. 2 представлена страница с перечнем прав администратора.

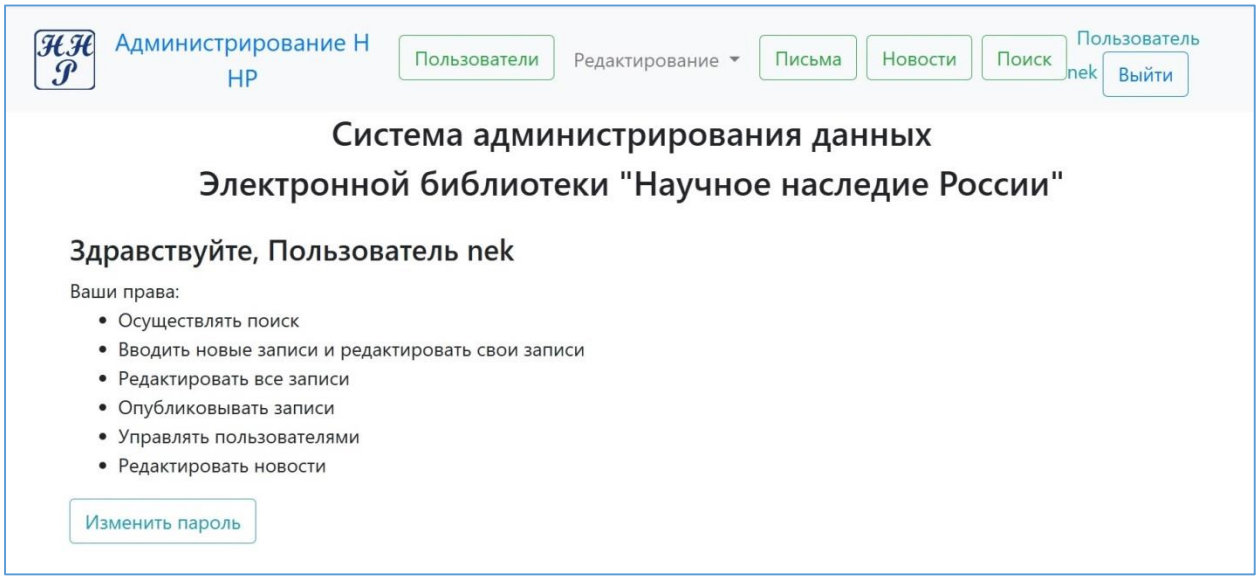


Рис. 2. Страница администратора ЭБ ННР.

Минимальные права пользователя включают только поиск и просмотр данных.

Пользователи, имеющие статус редактора, могут редактировать все записи; «рядовые» пользователи (операторы) могут вводить и редактировать только «свои записи»; редактировать новости имеют право только специальные сотрудники, они же работают с письмами внешних пользователей.

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС АДМИНИСТРАТИВНОГО БЛОКА

На рис. 2 в его верхней части представлен список опций, доступных администратору.

Редактирование. Эта опция включает возможности редактирования отдельных видов объектов (таких как персоны, публикации, музейные предметы), выбор вида объекта осуществляется из раскрывающегося списка. На рис. 3 представлена страница формирования запроса на поиск персон, рассматриваемых на предмет редактирования. В списке поисковых элементов представлены все элементы метаданных персоны, а также количество связанных с ней фотографий, внешних ссылок, публикаций и музейных объектов (количество задается в виде точного равенства или «меньше/больше»). Нижняя часть формы включает условия на текущий статус информации о персоне и условие на сортировку выбранных записей.

В примере, приведенном на рис. 3, выбираются персоны, ввод данных о которых завершен, а в биографии входит слово «Тибет».

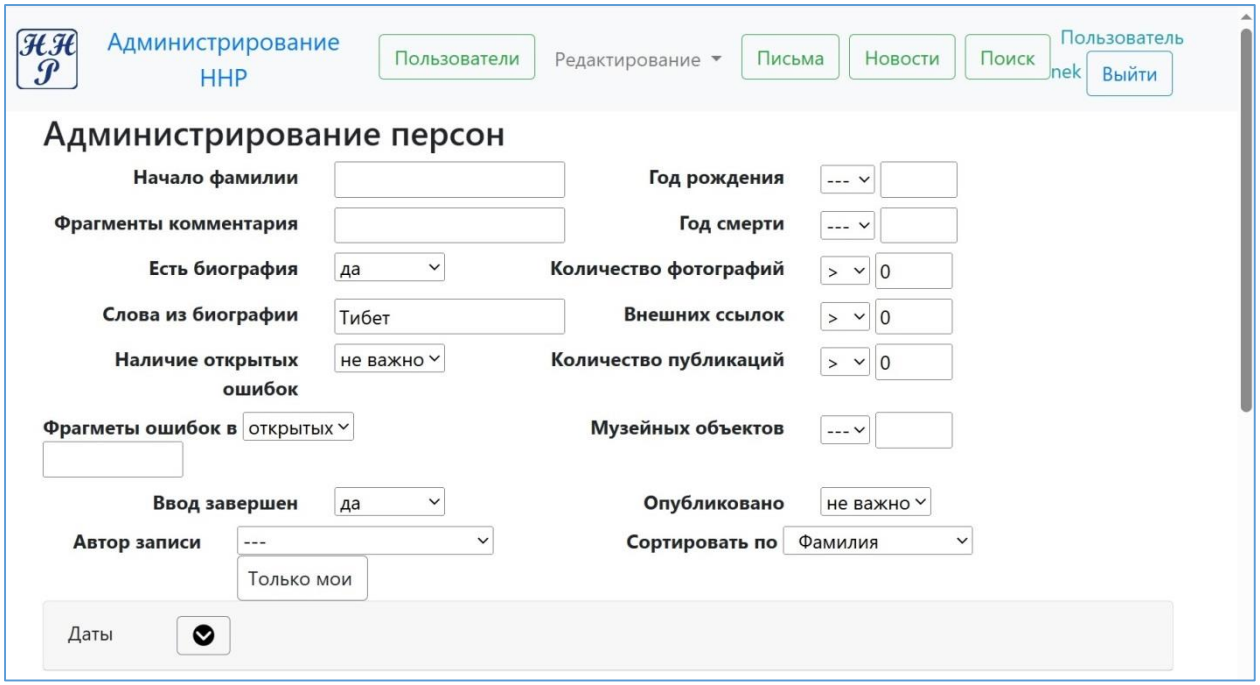


Рис. 3. Страницы запроса на выбор персоны для редактирования.

После ввода запроса система сообщает количество и краткий список выбранных записей (рис. 4).

Рис. 4. Страница выдачи результатов обработки запроса на редактирование персоны.

Каждую запись из списка можно раскрыть и получить список всех данных, относящихся к этой персоне. В качестве примера на рис. 5 представлена полная информация, касающаяся Гумбольдта. В верхней части списка содержатся данные о том, кто и когда ввел и последним редактировал инфор-

мацию. В правой части каждой строки имеется кнопка, после нажатия которой открывается окно для редактирования информации, относящейся к данному атрибуту. На рис. 6 представлена биография Гумбольдта с «подсвеченным» словом «Тибет», заданным в запросе.

Каждое значение атрибута после редактирования записывается в текущую базу данных и сразу же становится доступным для поиска и навигации.

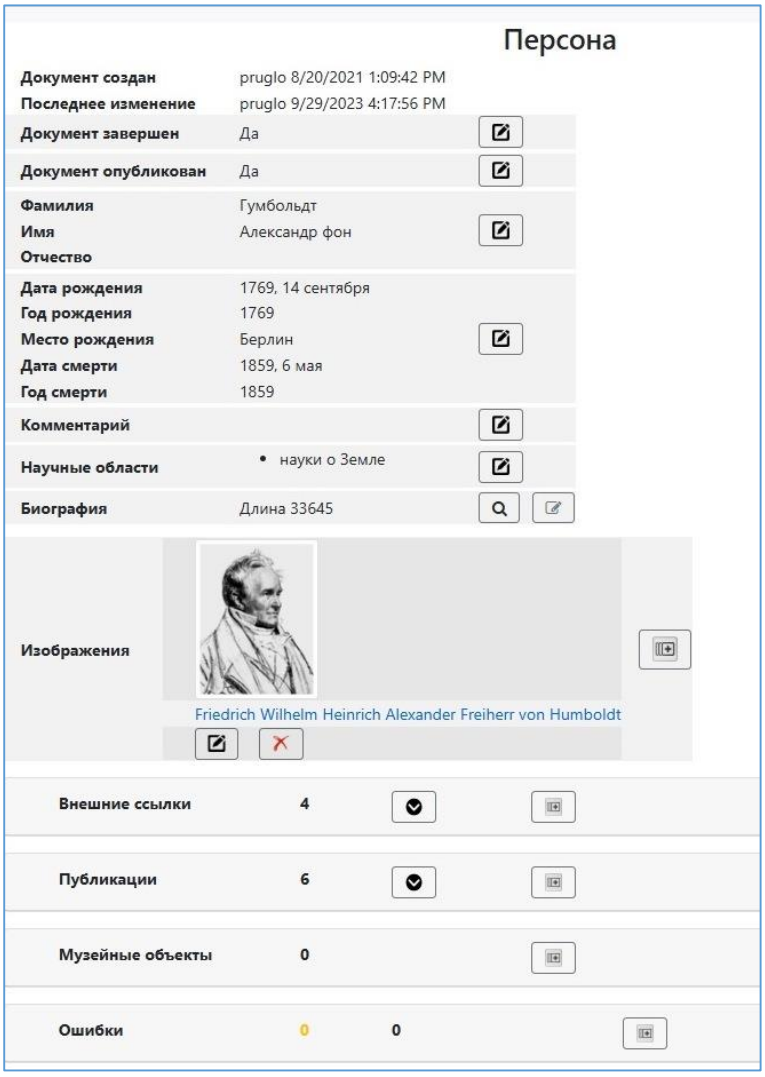


Рис. 5. Страница редактирования данных персоны.

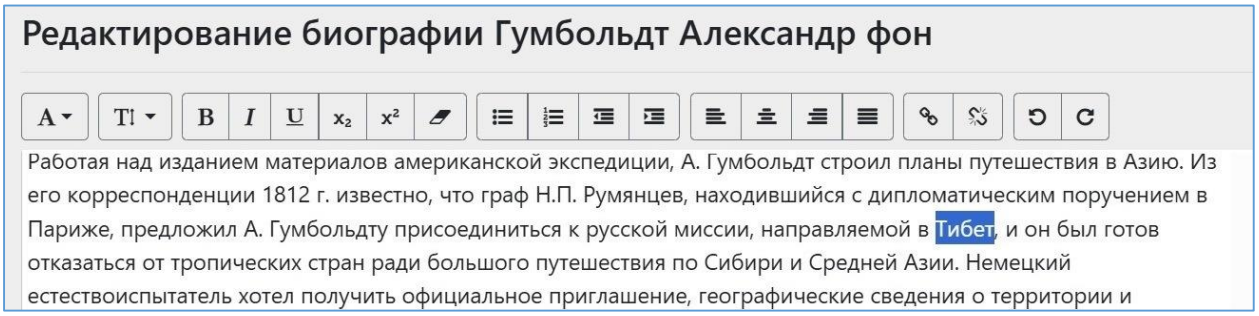


Рис. 6. Фрагмент страницы редактирования биографии персоны

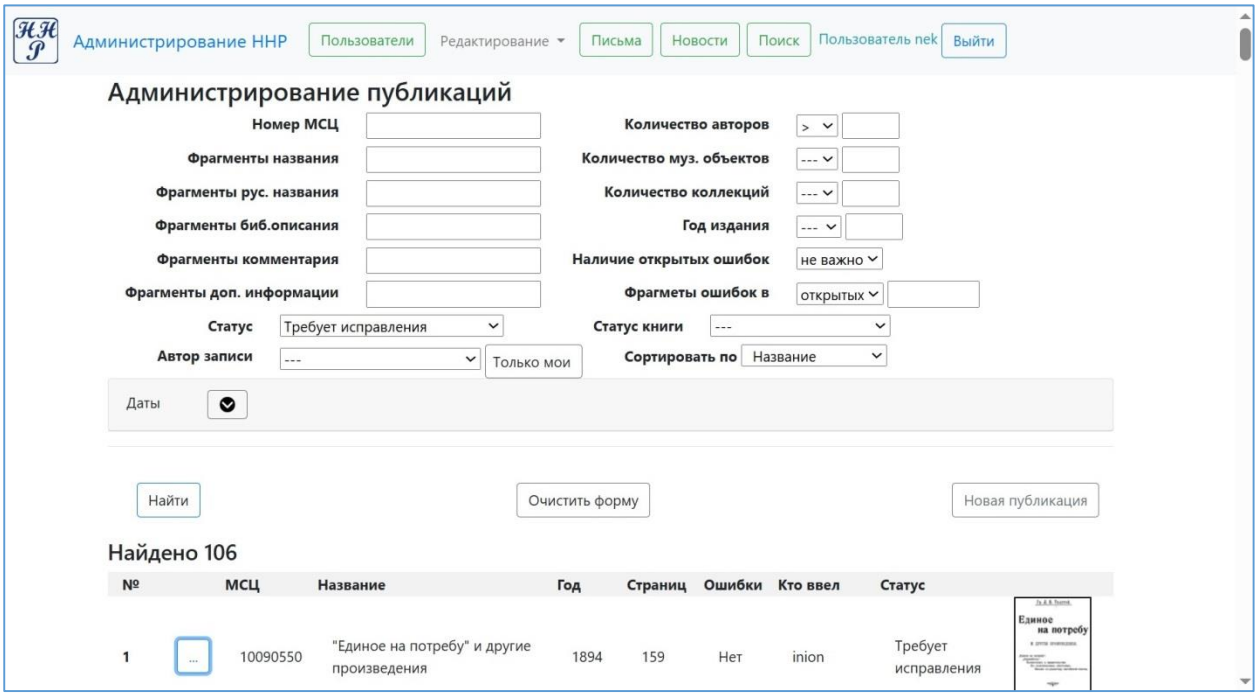


Рис. 7. Страница запроса на выбор публикации для редактирования.

На рис. 7 приведен пример администрирования публикаций. Запрос направлен на выявление публикаций, требующих исправления (в атрибуте «статус» внизу формы выбрано соответствующее значение). В результате обработки запроса был сформирован список из 106 публикаций. Открыв первую запись (Л. Н. Толстой, «Единое на потребу»), получаем список значений всех метаданных и информацию об ошибках (атрибут «comment» – «не читаются страницы 67–77, 130, 133») (рис. 8). Чтобы просмотреть страницы книги, нужно нажать на изображение титульной страницы (атрибут «электронная публикация»), предварительно выбрав размер показываемых страниц. На рис. 9 показан набор страниц размера «мелкий», при таком размере легко видеть брако-

ванные страницы. Нажав на изображение любой страницы, можно увеличить ее размер.

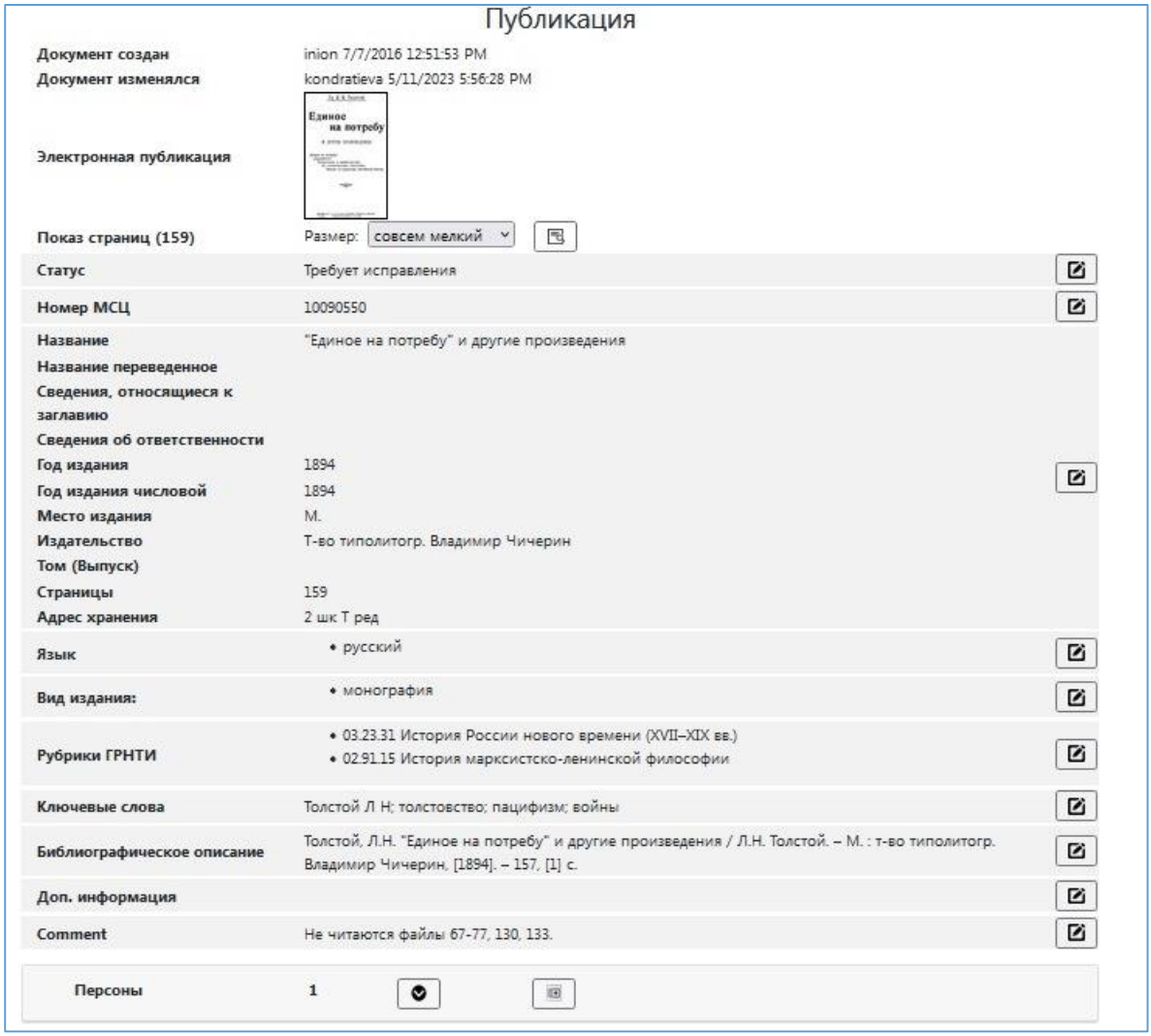


Рис. 8. Страница редактирования данных публикации.

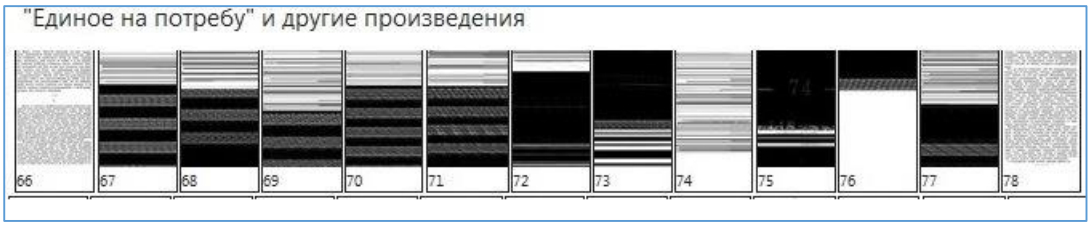


Рис. 9. Фрагмент выдачи бракованных страниц публикации.

В данном случае книга поступила из ИНИОН (атрибут «создан» – рис. 8), с которым необходимо решать вопрос об исправлении скана.

Опция экспорт данных формирует файл в формате *.csv и включает в каждую экспортируемую запись атрибуты, отмеченные в общем списке (рис. 10).

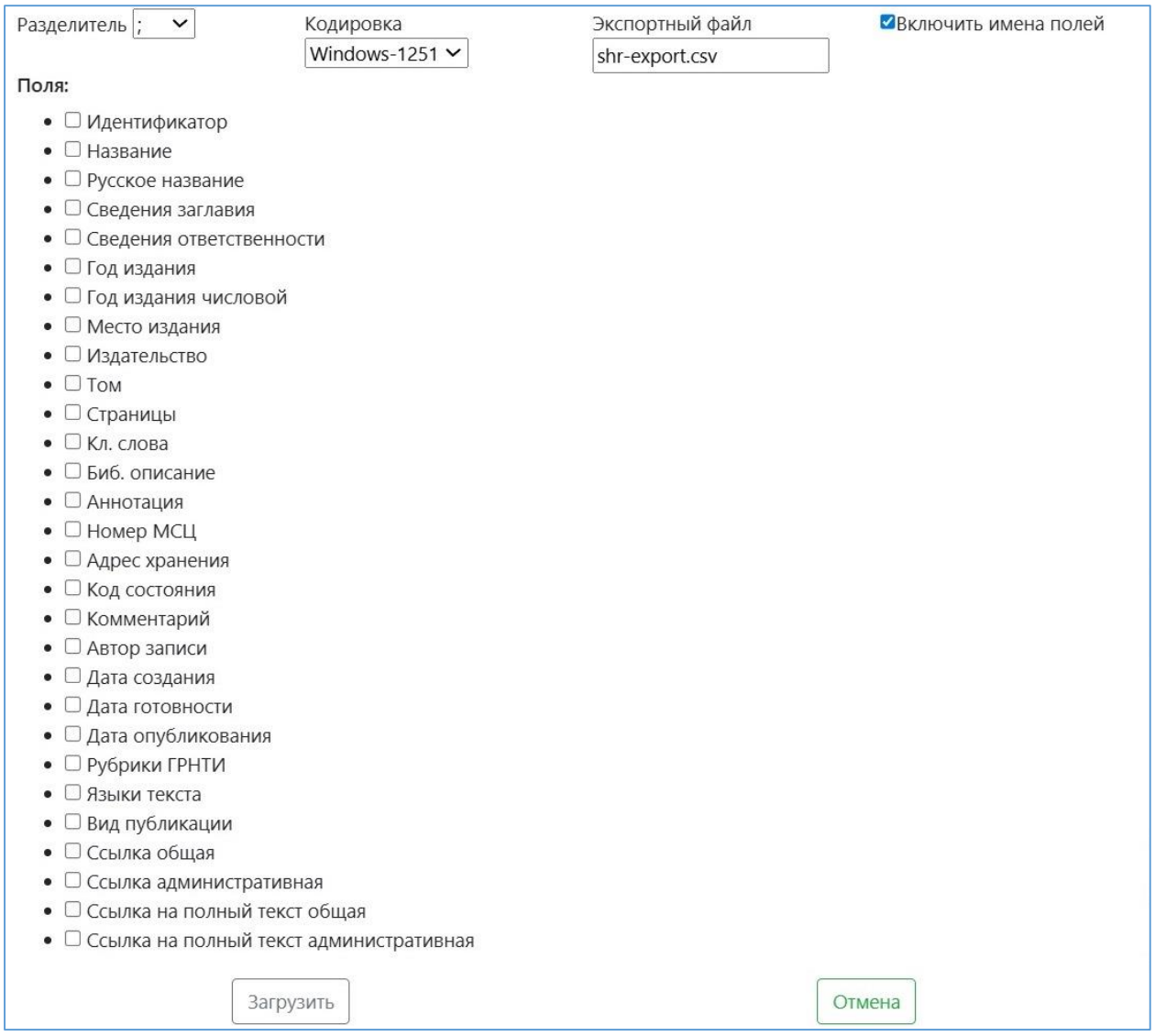


Рис. 10. Страница выбора атрибутов экспортируемых объектов.

ЗАКЛЮЧЕНИЕ

Как уже было сказано, ЭБ ННР развивается как элемент Единого цифрового пространства научных знаний. На накопленных массивах данных моделируются поисковые интерфейсы, сценарии формирования сложных запросов, алгоритмы навигации по связанным ресурсам. В этих направлениях развиваются

программные средства, поддерживающие электронную библиотеку в новом качестве.

Работы выполняются в рамках государственного задания НИЦ «Курчатовский институт».

СПИСОК ЛИТЕРАТУРЫ

1. *Каленов Н.Е., Савин Г.И., Серебряков В.А., Сотников А.Н.* Принципы построения и формирования электронной библиотеки «Научное наследие России» // Программные продукты и системы. 2012. Т. 4, № 100. С. 30–40.

2. *Погорелко К. П.* Динамика использования электронной библиотеки «Научное наследие России» // Информационное обеспечение науки: новые технологии: сб. науч. ст. М.: БЕН РАН, 2017. С. 192–200.

3. *Антопольский А.Б.* Научная информация и цифровое пространство знаний: постановка задачи для России // Наука и научная информация. 2020. Т. 3. № 1. С. 8–17.

4. *Савин Г.И.* Единое цифровое пространство научных знаний: цели и задачи // Информационные ресурсы России. 2020. № 5. С. 3–5. https://doi.org/10.51218/0204-3653-2020-5-3-5

5. *Каленов Н.Е., Погорелко К.П., Соболевская И.Н., Сотников А.Н.* Электронная библиотека «Научное наследие России» как элемент Единого цифрового пространства научных знаний // Научные и технические библиотеки. 2025. № 8. С. 101–123.

6. *Погорелко К.П.* Новая версия программного обеспечения электронной библиотеки «Научное наследие России» // Информационные ресурсы России. 2020. № 5. С. 27–29. https://doi.org/10.51218/0204-3653-2020-5-27-29

7. *Погорелко К.П.* О некоторых особенностях переноса Windows web приложений в среду Unix // Информационные процессы. 2023. Т. 23. № 3. С. 379–384. https://doi.org/10.53921/18195822_2023_23_3_379.

8. *Joe Reis, Matt Housle*. Fundamentals of Data Engineering. O'Reilly Media, Inc. 2022. 450p.

9. *К. Дж. Дейт.* Введение в системы баз данных; в 2 томах. Диалектика-Вильямс. 2024

10. *Кириллов С.А.* Технологическая платформа формирования цифровых ресурсов электронной библиотеки «Научное наследие России» // Информационные ресурсы России. 2020. № 5. С. 25–27.
https://doi.org/10.51218/0204-3653-2020-5-25-27

11. *Погорелко К.П., Савин Г.И.* Организация поиска в базе данных со связанными сущностями // Программные продукты и системы. 2024. № 4. С. 524–531. https://doi.org/10.15827/0236-235X.148.524-531. EDN ZDHAZG.

ADMINISTRATION OF THE SCIENTIFIC HERITAGE OF RUSSIA ELECTRONIC LIBRARY CONTENT

N. E. Kalenov[1] [0000-0001-5269-0988], K. P. Pogorelko[2] [0000-0002-4778-3060]

[1, 2]*National Research Center "Kurchatov Institute", Moscow, Russia*

[1]nekalenov@mail.ru, [2]konstpog@yandex.ru

Abstract

The electronic library "Scientific Heritage of Russia" (EL SHR) has been operating in open Internet access since 2010. The library integrates information about scientists who have contributed to the development of Russian science, their scientific publications, related archival materials, online resources and museum objects. The modern version of the NPR EB is developing as a model of a fragment of the Common Digital Space of Scientific Knowledge (CDSSK) and includes a number of functional blocks (metadata generation, publication of digitized documents and museum objects, organization of collections and exhibitions, content administration). The article describes the functionality of the electronic library's administrative block. The block is accessible to authorized users with the appropriate permissions. The block allows you to edit the metadata elements of each object type and the relationships between them, monitor the processing stages of specific objects entered in the electronic library, and export a specified set of related objects.

Keywords: *digital library, scientific heritage, management, search interface, related data.*

REFERENCES

1. *Kalenov N.E., Savin G.I., Serebryakov V.A., Sotnikov A.N.* Principy` postroeniya i formirovaniya e`lektronnoj biblioteki Nauchnoe nasledie Rossii // Programmny`e produkty` i sistemy. 2012. T. 4, № 100. S. 30–40.

2. *Pogorelko K.P.* Dinamika ispol`zovaniya e`lektronnoj biblioteki Nauchnoe nasledie Rossii // Informacionnoe obespechenie nauki: novy`e texnologii: sb. nauch. st. M.: BEN RAN. 2017. S. 192–200.

3. *Antopol`skij A.B.* Nauchnaya informaciya i cifrovoe prostranstvo znanij: postanovka zadachi dlya Rossii // Nauka i nauchnaya informaciya. 2020. T. 3. № 1. S. 8–17.

4. *Savin G.I.* Edinoe cifrovoe prostranstvo nauchny`x znanij: celi i zadachi // Informacionny`e resursy` Rossii. 2020. № 5. S. 3–5. https://doi.org/10.51218/0204-3653-2020-5-3-5

5. *Kalenov N.E., Pogorelko K.P., Sobolevskaya I.N., Sotnikov A.N.* E`lektronnaya biblioteka “Nauchnoe nasledie Rossii” kak e`lement Edinogo cifrovogo prostranstva nauchny`x znanij // Nauchny`e i texnicheskie biblioteki. 2025. № 8. S. 101–123.

6. *Pogorelko K.P.* Novaya versiya programmnogo obespecheniya e`lektronnoj biblioteki «Nauchnoe nasledie Rossii» // Informacionny`e resursy` Rossii. 2020. № 5. S. 27–29. https://doi.org/10.51218/0204-3653-2020-5-27-29

7. *Pogorelko K.P.* O nekotory`x osobennostyax perenosa Windows web prilozhenij v sredu Unix // Informacionny`e processy. 2023. T. 23. № 3. S. 379–384. https://doi.org/10.53921/18195822_2023_23_3_379

8. *Joe Reis, Matt Housle*. Fundamentals of Data Engineering. O’Reilly Media, Inc. 2022. 450 p.

9. *K. Dzh. Dejt.* Vvedenie v sistemy` baz danny`x; v 2 tomax. Dialektika-Vil`yams. 2024

10. *Kirillov S.A.* Texnologicheskaya platforma formirovaniya cifrovy`x resursov e`lektronnoj biblioteki “Nauchnoe nasledie Rossii” // Informacionny`e resursy` Rossii. 2020. № 5. S. 25–27. https://doi.org/10.51218/0204-3653-2020-5-25-27

11. *Pogorelko K.P., Savin G.I.* Organizaciya poiska v baze danny`x so svyazanny`mi sushhnostyami // Programmny`e produkty` i sistemy`. 2024. № 4. S. 524–531. https://doi.org/10.15827/0236-235X.148.524-531

СВЕДЕНИЯ ОБ АВТОРАХ

КАЛЕНОВ Николай Евгеньевич – главный научный сотрудник отделения суперкомпьютерных систем и параллельных вычислений Национального исследовательского центра «Курчатовский институт», доктор технических наук, профессор.

Nikolay Evgenievich KALENOV – chief researcher at the Department of Supercomputer Systems and Parallel Computing at the National Research Center "Kurchatov Institute", Doctor of Technical Sciences, Professor.

email: nekalenov@mail.ru

ORCID: 0000-0001-5269-0988

ПОГОРЕЛКО Константин Павлович – старший научный сотрудник отделения суперкомпьютерных систем и параллельных вычислений Национального исследовательского центра «Курчатовский институт», доктор технических наук, профессор.

Konstantin Pavlovich POGORELKO – senior researcher at the Department of Supercomputer Systems and Parallel Computing at the National Research Center "Kurchatov Institute", Doctor of Technical Sciences, Professor.

email: konstpog@yandex.ru

ORCID: 0000-0002-4778-3060

Материал поступил в редакцию 27 февраля 2026 года

УДК 013+004.65

О ВКЛЮЧЕНИИ МУЗЕЙНЫХ ОБЪЕКТОВ В ЕДИНОЕ ЦИФРОВОЕ ПРОСТРАНСТВО НАУЧНЫХ ЗНАНИЙ

С. А. Кириллов[1] [0000-0001-7560-0041], И. Н. Соболевская[2] [0000-0002-9461-3750]

[1, 2]*Национальный исследовательский центр «Курчатовский институт», г. Москва, Россия*

[1]skirillov@jscc.ru, [2]nik_first@mail.ru

Аннотация

Работа посвящена вопросам интеграции музейных объектов в Единое цифровое пространство научных знаний (ЕЦПНЗ). Рассмотрена эволюция музейного предмета от изолированного артефакта до «интеллектуального интерфейса» – связанного элемента сети знаний. Описана технология оцифровки трехмерных музейных объектов с помощью spin-съемки. На примере коллекции муляжей грибов Государственного биологического музея продемонстрирован процесс включения объектов в ЕЦПНЗ с использованием структурированных данных и интерактивных 3D-моделей. Работа выполнена в рамках государственного задания и демонстрирует потенциал ЕЦПНЗ как универсальной среды для сохранения и распространения научного наследия.

Ключевые слова*: Единое цифровое пространство научных знаний, интеграция музейных объектов, spin-анимация, онтология, 3D-объект, воксель, облако точек.*

ВВЕДЕНИЕ

Распространение научных знаний начиная с древних времен осуществлялось в письменном виде (рукописные и печатные книги, статьи, архивные материалы), в виде материальных предметов, хранящихся в музеях, и кинодокументов. Все эти источники в последние десятилетия активно переводятся в цифровую форму, что позволяет практически неограниченно и мгновенно распространять накопленные знания.

Единое цифровое пространство научных знаний (ЕЦПНЗ) – это среда, обеспечивающая хранение и распространение научных знаний, содержащихся изначально в перечисленных выше источниках [1–4].

Одним из направлений работ, связанных с проектированием ЕЦПНЗ, является разработка методологии отражения в пространстве музейных объектов. Если технология перевода в цифровую форму и визуализации «плоских» музейных предметов (рисунков, фотографий, документов и пр.) разработана и широко используется без особых проблем, то вопрос перевода в цифровую форму и отражения с необходимой точностью трехмерных музейных предметов является серьезной научной проблемой [5–8].

В ЕЦПНЗ музейный объект эволюционирует от «вещи в себе» к «интеллектуальному интерфейсу». Он превращается в связанную единицу информации. А именно: чтобы получить информацию об экспонате, находящемся в музее, надо прийти в музей, найти экспонат, прочитать текст о нем, при этом связи этого объекта, например, с историческим контекстом, людьми, событиями и другими предметами неочевидны и скрыты. И это при условии, что объект экспонируется, а не находится в фондохранилищах музея. За пределами музейных стен и каталогов для узкого круга специалистов этот объект практически не существует. В свою очередь, для цифрового образа музейного объекта ценностью является не только его «содержимое», но и сетевые связи между всеми элементами пространства знаний. Таким образом, оцифрованный музейный объект, интегрированный в ЕЦПНЗ, может быть «описан» структурированными данными (полями «автор», «дата создания», «материал», «техника», «географическая привязка», «историческое событие» и т. д.) и «приобретает» связи с другими объектами, т. е. становится «интеллектуальным интерфейсом»: пользователь взаимодействует не с изолированным предметом, а с сетью знаний, точкой входа в которую является цифровой образ музейного объекта.

1. ИНТЕГРАЦИЯ МУЗЕЙНЫХ ОБЪЕКТОВ В ЕДИНОЕ ЦИФРОВОЕ ПРОСТРАНСТВО НАУЧНЫХ ЗНАНИЙ

Принципы интеграции музейных объектов в ЕЦПНЗ направлены на то, чтобы превратить физический артефакт в «интеллектуальный интерфейс» и связанную единицу информации.

Для интеграции в ЕЦПНЗ цифровые образы музейных объектов должны соответствовать определенным требованиям: многоуровневость представления (стандартизированное, структурированное описание атрибутов – основа для связывания и поиска) и визуальный ряд (одно или серия изображений высокого разрешения; 3D-модели должны быть в форматах, поддерживающих веб-визуализацию).

Таким образом, в ЕЦПНЗ музейный объект превращается в «интеллектуальный интерфейс», позволяющий не только просматривать статические данные, но и исследовать связи объектов в их динамике и трехмерные образы объектов [9].

2. ПРЕДСТАВЛЕНИЕ 3D-ОБЪЕКТОВ В ЕДИНОМ ЦИФРОВОМ ПРОСТРАНСТВЕ НАУЧНЫХ ЗНАНИЙ

Интеграция трехмерных музейных объектов в ЕЦПНЗ требует применения специализированных технологий, адаптированных к задачам ЕЦПНЗ. А именно: технология должна давать результат, пригодный для исследований, а не только для визуализации; данные должны легко интегрироваться с другими системами (архивами, библиотеками, исследовательскими базами).

Один из самых распространенных и первых методов в трехмерном моделировании – это полигональное моделирование [10, 11]. Этот метод заключается в создании объемных моделей объектов с помощью полигонов – многоугольных элементов, которые формируют поверхность модели. Однако для формирования и представления 3D-моделей музейных предметов этот метод не эффективен по следующим причинам [12]:

- полигональное моделирование – это интерпретация. Художник-модельер вручную создает форму, опираясь на фотографии или наброски. Оцифровка музейных предметов должна быть документом, а не иллюстрацией;
- нерентабельность для музейных объектов: ручное создание точной модели сложного трехмерного объекта (окаменелость, скелет, керамика и т. п.) требует сотен часов работы высококвалифицированного специалиста;
- невозможность массовой оцифровки: метод не подходит для создания больших цифровых коллекций (тысячи и десятки тысяч экспонатов) из-за огромных затрат времени и средств;

- в саму полигональную сетку сложно внедрить семантические слои – точечно привязать метаданные к конкретным вершинам или полигонам (например, пометить область реставрации или конкретную трещину).

Поэтому для формирования цифровых 3D-объектов используют методы, альтернативные полигональному моделированию: воксели (воксель – объемный пиксель), облака точек, неявные функции, параметрические поверхности, а также spin-анимацию [13–17]. Существует несколько ключевых методов представления 3D-объектов, которые не используют полигональные модели. Каждый из них имеет свои уникальные характеристики, преимущества и недостатки, что делает их пригодными для различных типов объектов и приложений. Каждый из этих методов применяется в зависимости от типа объекта, требуемой детализации и наличия необходимых вычислительных мощностей.

В интерактивных цифровых 3D-моделях важна оптимизация для достижения баланса между визуальным качеством, производительностью (частотой кадров) и ресурсозатратами (памятью, вычислительной мощностью) при создании и воспроизведении последовательности кадров, имитирующих, например, вращение объекта [18]. Под интерактивной цифровой 3D-моделью понимается взаимодействие пользователя с таким объектом, в том числе вращение, масштабирование (приближение/отдаление), панорамирование, выбор ракурса для детального изучения, анализа объекта и т. п.

2.1. Воксели

Воксели представляют собой трехмерные эквиваленты пикселей, где каждый воксель является кубическим элементом в трехмерной сетке, содержащим информацию об объеме и цвете. В отличие от плоских пикселей, воксели обладают шириной, высотой и глубиной, что позволяет им представлять объемные объекты. Комбинируя множество окрашенных вокселей, можно создавать сложные 3D-формы, при этом увеличение количества вокселей приводит к повышению разрешения и детализации [19].

Одним из основных недостатков является ограниченный уровень детализации по сравнению с полигональным моделированием, поскольку воксели по своей природе являются блочными и могут испытывать трудности с представле-

нием гладких поверхностей или мелких деталей. Рендеринг векселей (метод визуализации трехмерных объектов и сцен, при котором базовым элементом представления данных является не полигон – треугольник, а вексель) может быть вычислительно затратным, особенно при высоком разрешении [20–22].

2.2. Облака точек

Облако точек представляет собой набор отдельных точек данных в трехмерном пространстве, каждая из которых имеет свои координаты *XYZ* и может содержать дополнительную информацию, такую как цвет, нормали и временные метки. Облака точек обычно создаются с помощью 3D-сканеров или программного обеспечения для фотограмметрии, которые измеряют множество точек на внешних поверхностях объектов [23, 24].

Несмотря на свою точность и детализацию, облака точек имеют некоторые недостатки, например, отсутствие связности между отдельными точками затрудняет визуализацию и анализ объекта. Большие наборы данных облаков точек требуют значительной вычислительной мощности для обработки. Для рендеринга (процесса синтеза изображения из данных) и анимации облака точек часто преобразуются в полигональные сетки с использованием методов реконструкции поверхности [25].

2.3. Неявные функции

Неявные функции определяют поверхность как множество точек (*x, y, z*), для которых значение функции F(*x, y, z*) = 0. В отличие от явных или параметрических представлений, неявные поверхности определяются условием, которому должны удовлетворять точки на поверхности. Примером неявных поверхностей являются сферы.

Визуализация неявных поверхностей является сложной задачей, которая часто требует использования таких методов, как трассировка лучей или алгоритм марширующих кубов для преобразования неявной функции в полигональную сетку для рендеринга. Оценка неявной функции для каждой точки может быть вычислительно затратной, особенно для сложных функций [26, 27].

2.4. Параметрические поверхности

Параметрические поверхности определяются набором уравнений, которые выражают координаты точек на поверхности как функции двух параметров. Примерами параметрических поверхностей являются сплайновые поверхности, NURBS (неравномерные рациональные B-сплайны) и поверхности вращения.

Параметрические поверхности обеспечивают точный контроль формы поверхности и широко используются в системах автоматизированного проектирования и промышленном дизайне. Они хорошо подходят для представления гладких криволинейных поверхностей и облегчают генерацию точек на поверхности.

По сравнению с неявными поверхностями, параметрические поверхности могут иметь топологические ограничения. Вычисление пересечений (луч – поверхность, поверхность – поверхность) может быть сложным [28, 29].

2.5. Технология spin-съемки

Для представления 3D-музейных объектов в ЕЦПНЗ используется технология спин-съемки (или круговой 360-ной° съемки) [30]. Такая съемка создает структурированный набор изображений, из которого можно:

- сформировать интерактивную анимацию для представления музейного объекта в цифровом формате;
- сгенерировать полноценную 3D-модель с помощью алгоритмов фотограмметрии;
- извлечь высокодетализированные текстуры для исследований.

Для ЕЦПНЗ технология спин-анимации является эффективной, поскольку что она эффективно преобразует физический музейный предмет в структурированный, многоразовый, машиночитаемый и визуально полный цифровой актив, который служит универсальным фундаментом для научных исследований, образования и публичного представления в рамках единой стандартизированной системы [31].

Таким образом, один раз проведенная качественная спин-съемка становится долгосрочным цифровым активом, который можно использовать для различных целей в рамках ЕЦПНЗ.

3. ПРИМЕР ВИРТУАЛЬНОЙ 3D-КОЛЛЕКЦИИ МУЗЕЙНЫХ ПРЕДМЕТОВ КАК ЭЛЕМЕНТА ЕДИНОГО ЦИФРОВОГО ПРОСТРАНСТВА НАУЧНЫХ ЗНАНИЙ

В Государственном биологическом музее имени К. А. Тимирязева (ГБМТ) находится уникальная коллекция муляжей грибов (более 200 видов) в имитации природной обстановки. Каждый музейный предмет представляют собой палету, на которой установлены макеты грибов, изготовленные из папье-маше, и находящиеся в «природной среде», выполненной из природных материалов (мха, травы, листьев и т. п.) (рис. 1). По мнению сотрудников ГБМТ, эта коллекция представляет интерес как для обычных посетителей, так и для специалистов-микологов. Поэтому было принято решение сформировать и интегрировать в ЕЦПНЗ цифровую коллекцию на основе таких муляжей, находящихся как в основной экспозиции, так и в запасниках ГБМТ.

Рис. 1. Пример цифрового образа музейного предмета «Подгруздь белый».

Каждый элемент виртуальной коллекции, созданной авторами, – это 3D-образ исходного объекта. Для визуализации 3D-моделей и представления их в цифровом формате используется технология интерактивной анимации [9, 32–34].

В большинстве отечественных музеев, в том числе и в ГБМТ, используется комплексная автоматизированная музейная информационная система «КАМИС» [35, 36]. Она содержит структурированную информацию о музейных объектах и аппарат для ее экспорта.

С помощью алгоритмов формирования контента конкретного подпространства ЕЦПНЗ, реализованных в модельном программном комплексе, описанном в [36], осуществляется процесс загрузки данных о музейных объектах в ЕЦПНЗ.

На первом этапе включения коллекции грибов в ЕЦПНЗ с помощью унифицированной диалоговой программы [37] создается подпространство «Биология». В нем формируются справочники следующих классов: «Грибы», «Объекты биологических музеев», «Биологические коллекции», «Изображения и мультимедиа в биологии». Для каждого из сформированных классов созданы справочники соответствующих атрибутов.

В результате работы программы загрузки формируются все необходимые словари объектов и связей. Таким образом, все связанные данные оказываются загруженными в структуру ЕЦПНЗ. Существующие в настоящее время программные средства ЕЦПНЗ позволяют просматривать объекты любого класса и осуществлять навигацию по их связям.

Названия грибов являются активными ссылками, которые обеспечивают переход к информации о выбранном объекте и его связях. Связи объекта «муляж гриба “Груздь черный”» показывают, что он хранится в Государственном биологическом музее и входит в состав микологической коллекции этого музея. Ссылка связи «модель гриба “Груздь черный”» обеспечивает переход на 3D-модель данного объекта [38].

ЗАКЛЮЧЕНИЕ

Процесс включения музейных объектов в ЕЦПНЗ представляет собой комплексную задачу, требующую сочетания технологических решений, стандартизации данных и методологической проработки, что в итоге ведет к созданию универсальной, открытой и многофункциональной среды для сохранения и распространения научных знаний.

Концепция представления музейных объектов в ЕЦПНЗ предполагает интеграцию в единую систему знаний целостной информационной модели, что позволяет перейти от простого представления отдельных оцифрованных копий музейных предметов к построению сложных информационных моделей, где каждый музейный объект занимает определенное место в онтологии ЕЦПНЗ.

Благодарности

Работа выполнена в рамках государственного задания НИЦ «Курчатовский институт».

СПИСОК ЛИТЕРАТУРЫ

1. *Савин Г.И.* Единое цифровое пространство научных знаний: цели и задачи // Информационные ресурсы России. 2020. № 5. С. 3–5.

2. ISO 25964 the international standard for thesauri and interoperability with other vocabularies. https://www.niso.org/schemas/iso25964

3. *Каленов Н.Е., Сотников А.Н.* Единое цифровое пространство научных знаний как интегратор политематических информационных ресурсов // Доклады Российской академии наук. Математика, информатика, процессы управления. 2024. Т. 515. С. 114–123. https://doi.org/10.31857/S2686954324010177

4. *Антопольский А.Б., Босов А.В., Савин Г.И., Сотников А.Н., Цветкова В.А., Каленов Н.Е., Серебряков В.А., Ефременко Д.В.* Принципы построения и структура единого цифрового пространства научных знаний (ЕЦПНЗ) // Научно-техническая информация. Сер. 1. 2020. № 4. С. 9–17.

5. *Каленов Н.Е., Сотников А.Н.* Архитектура единого цифрового пространства научных знаний // Информационные ресурсы России. 2020. № 5. С. 5–8.

6. *Vlasova S.A., Kalenov N.E., Kirillov S.A., Sobolevskaya I.N., Sotnikov A.N.* Modeling of a fragment of the Common digital space of scientific knowledge by the example of museum collections // Scientific and Technical Information Processing, 2025. Vol. 52, No. 2. P. 129–134. https://doi.org/10.3103/S014768822570011X

7. *Каленов Н.Е., Соболевская И.Н., Сотников А.Н.* Виртуальная выставка как элемент популяризации научных знаний // Научные и технические библиотеки. 2024. № 2. С. 107–122.

8. *Власова С.А., Каленов Н.Е., Кириллов С.А., Соболевская И.Н., Сотников А.Н.* Естественнонаучные коллекции как элемент Единого Цифрового Пространства Научных Знаний // Научный сервис в сети Интернет: труды XXVI Всероссийской научной конференции. М.: ИПМ им. М.В. Келдыша. 2024. С. 39–49.

9. *Paszkowska M. et al.* 3D technologies for Intangible Cultural Heritage Preservation—Literature Review for selected databases. Heritage Science [Электронный ресурс]. 2022. Vol. 10(1). No. 3.
URL: https://link.springer.com/article/10.1186/s40494-021-00633-x (Дата обращения: 07.02.2026).

10. *Кириллов С.А., Соболевская И.Н., Сотников А.Н.* Принципы формирования и представления междисциплинарных коллекций в цифровом пространстве научных знаний // Электронные библиотеки. 2021. Т. 24. № 2. С. 294–314.

11. *Botsch M., Pauly M., Kobbelt L., Alliez P., Lévy B., Bischoff S., Röoss C.* Geometric modeling based on polygonal meshes Video files associated with this course are available from the citation page // SIGGRAPH '07: ACM SIGGRAPH 2007 courses. P. 49. https://doi.org/10.1145/1281500.1281640

12. *Russo M.* Polygonal modeling: basic and advanced techniques. Jones & Bartlett Learning. 2006. 411 p.

13. *Ju T.* Fixing geometric errors on polygonal models: a survey //Journal of Computer Science and Technology. 2009. Vol. 24.No. 1. P. 19–29.

14. *Chuvikov D.A. et al.* 3D modeling and 3D objects creation technology analysis for various intelligent systems // International Journal of Advanced Studies. 2014. Vol. 4. No. 4. P. 16–22.

15. *Beraldin J. A. et al.* Real world modelling through high resolution digital 3D imaging of objects and structures // ISPRS Journal of Photogrammetry and Remote Sensing. 2000. Vol. 55. No. 4. P. 230–250.

16. *Palka D., Sobota M., Buchwald P.* 3D object digitization devices in manufacturing engineering applications and services // Multidisciplinary Aspects of Production Engineering. 2020. Vol. 3. P. 450–463.

17. *Yan Y., Letscher D., Ju T.* Voxel cores: Efficient, robust, and provably good approximation of 3d medial axes // ACM Transactions on Graphics (TOG). 2018. Vol. 37. No. 4. P. 1–13.

18. *Koeva M.N.* 3D modelling and interactive web-based visualization of cultural heritage objects // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2016. Vol. 41. P. 297–303.

19. Zhengren Wang 3D Representation Methods: A Survey URL: https://arxiv.org/html/2410.06475v1 (Дата обращения: 16.02.2026)

20. Spatial Representation URL: https://app.uxcel.com/courses/3d-design-foundations/d-spatial-representation-514 (Дата обращения: 16.02.2026)

21. *Chen C., Chen Z., Zhang J., Tao D*. SASA: Semantics-Augmented Set Abstraction for Point-Based 3D Object Detection // Proceedings of the AAAI Conference on Artificial Intelligence. 2022. Vol. 36, No. 1. P. 221–229. https://doi.org/10.1609/aaai.v36i1.1989

22. *Wang W. et al.* Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction // arXiv preprint arXiv:2509.19297. 2025.

23. *Cao R. et al.* Poxel: Voxel Reconstruction for 3D Printing // arXiv preprint arXiv:2501.10474. 2025.

24. What is the Difference Between 3D Point Clouds vs. 3D BIM Models? URL: https://www.existingconditions.com/knowledge-center-articles/3d-point-cloud-vs-model

25. *Yu Y. et al.* From comparison to integration: A workflow evaluation of 3D Gaussian splatting and LiDAR point cloud for modern architectural heritage // Automation in Construction. 2025. Vol. 180. No. 106509.

26. *Kwon O., Yu J.* Realistic and Interactive Virtual Museum Representation Using 3D Gaussian Splatting // ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2025. P. 185–192.

27. *Chibane J., Alldieck T., Pons-Moll G.* Implicit functions in feature space for 3d shape reconstruction and completion // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. P. 6970–6981.

28. *Bhatnagar B.L. et al.* Combining implicit function learning and parametric models for 3d human reconstruction // European conference on computer vision. Cham: Springer International Publishing, 2020. P. 311–329.

29. *Bison G. et al.* Automatizing 3D reconstruction pipelines for speeding-up cultural heritage digitization // Multimedia Tools and Applications. 2025. P. 1–21.

30. *Kilis N. et al.* AI tools for generating Digital Heritage Twins enhancing storytelling in educational games // Digital Applications in Archaeology and Cultural Heritage. 2025. e00451.

31. *Vargün Ö.* Visual analysis of 3D characters and animations used in the presentation of cultural heritage at museums // Journal of Arts. 2023. Vol. 6. No. 2. P. 141–155.

32. *Sobolevskaya I.N., Sotnikov A.N.* Virtual Exhibition as a Means of Integrating into a Unified Digital Space of Scientific Knowledge the Information Systems in the Field of Science and Culture // Automatic Documentation and Mathematical Linguistics, 2025. Vol. 58. S43–S50. https://doi.org/10.3103/S0005105525700098

33. *Каленов Н.Е., Погорелко К.П., Соболевская И.Н., Сотников А.Н.* Электронная библиотека «Научное наследие России» как элемент Единого цифрового пространства научных знаний // Научные и технические библиотеки. 2025. № 8. С. 101–123. https://doi.org/10.33186/1027-3689-2025-8-101-123

34. *Skublewska-Paszkowska M. et al.* 3D technologies for Intangible Cultural Heritage Preservation—Literature Review for selected databases. Heritage Science [Электронный ресурс]. 2022. Vol. 10(1). No. 3.
URL: https://link.springer.com/article/10.1186/s40494-021-00633-x (Дата обращения: 10.02.2026).

35. Круглый стол «Культурная вакцина. Музей в цифровой реальности» на Петербургском международном юридическом форуме – 2021 URL: https://www.hermitagemuseum.org/news/news_107_21?lng=el (Дата обращения: 16.02.2026)

36. *Юрина Ю.Г., Шалыганова О.С.* КАМИС-2000 в музее В.Г. Белинского // В сборнике: Современное общество, образование и наука. сборник научных трудов по материалам Международной научно-практической конференции: в 16 частях. 2015. С. 143–145.

37. *Румянцев М.С.* Использование информационных технологий в научно-фондовой деятельности музея (на примере программы «КАМИС») // В сборнике: XVI ежегодная научная сессия аспирантов и молодых ученых. материалы Всероссийской научной конференции. Вологда, 2023. С. 303–305.

38. *Власова С.А., Каленов Н.Е.* Диалоговый программный комплекс формирования онтологии Единого цифрового пространства научных знаний // Программные продукты и системы. 2024. Т. 37, № 4. С. 514–523.

39. *Власова С.А., Каленов Н.Е., Кириллов С.А., Соболевская И.Н., Сотников А.Н.* Моделирование фрагмента Единого цифрового пространства научных

знаний на примере музейных коллекций // Научно-техническая информация. Серия 1. 2025. № 5. С. 30–35. https://doi.org/10.36535/0548-0019-2025-05-4

40. Грибное пространство URL: http://exibitions.jscc.ru/360/15619/ (Дата обращения: 16.02.2026)

ON THE INTEGRATION OF MUSEUM OBJECTS INTO THE COMMON DIGITAL SPACE OF SCIENTIFIC KNOWLEDGE

S. A. Kirillov[1] [0000-0001-7560-0041], I. N. Sobolevskaya[2] [0000-0002-9461-3750

[1, 2]*National Research Centre "Kurchatov Institute", Moscow, Russia*

[1]skirillov@jscc.ru, [2]nik_first@mail.ru

Abstract

The work addresses the issues of integrating museum objects into the Common Digital Space of Scientific Knowledge (CDSSK). It examines the evolution of a museum item from an isolated artifact to an "intelligent interface" – a linked element of a knowledge network. The technology for digitizing three-dimensional museum objects using spin-scanning is described. Using the collection of mushroom models from the State Biological Museum as an example, the process of incorporating objects into the CDSSK using structured data and interactive 3D models is demonstrated. The work is carried out within the framework of a state assignment and demonstrates the potential of the CDSSK as a universal environment for preserving and disseminating scientific heritage.

Keywords*: Common Digital Space of Scientific Knowledge, integration of museum objects, spin animation, ontology, 3D object, voxel, point cloud.*

REFERENCES

1. *Savin G.I.* Edinoe cifrovoe prostranstvo nauchnykh znanij: celi i zadachi // Informacionnye resursy Rossii. 2020. № 5. S. 3–5.

2. ISO 25964 the international standard for thesauri and interoperability with other vocabularies. https://www.niso.org/schemas/iso25964

3. *Kalenov N.E., Sotnikov A.N.* Edinoe cifrovoe prostranstvo nauchnykh znanij kak integrator politematicheskikh informacionnykh resursov // Doklady Rossijskoj akademii nauk. Matematika, informatika, processy upravleniya. 2024. T. 515. S. 114–123. https://doi.org/10.31857/S2686954324010177

4. *Antopol'skij A.B., Bosov A.V., Savin G.I., Sotnikov A.N., Cvetkova V.A., Kalenov N.E., Serebryakov V.A., Efremenko D.V.* Principy postroeniya i struktura edinogo cifrovogo prostranstva nauchnykh znanij (ECPNZ) // Nauchno-tekhnicheskaya informaciya. Ser. 1. 2020. № 4. S. 9–17.

5. *Kalenov N.E., Sotnikov A.N.* Arkhitektura edinogo cifrovogo prostranstva nauchnykh znanij // Informacionnye resursy Rossii. 2020. № 5. S. 5–8.

6. *Vlasova S.A., Kalenov N.E., Kirillov S.A., Sobolevskaya I.N., Sotnikov A.N.* Modeling of a fragment of the Common digital space of scientific knowledge by the example of museum collections // Scientific and Technical Information Processing. 2025. Vol. 52, No. 2. P. 129–134. https://doi.org/10.3103/S014768822570011X

7. *Kalenov N.E., Sobolevskaya I.N., Sotnikov A.N.* Virtual'naya vystavka kak ehlement populyarizacii nauchnykh znanij // Nauchnye i tekhnicheskie biblioteki. 2024. № 2. S. 107–122.

8. *Vlasova S.A., Kalenov N.E., Kirillov S.A., Sobolevskaya I.N., Sotnikov A.N.* Estestvennonauchnye kollekcii kak ehlement Edinogo Cifrovogo Prostranstva Nauchnykh Znanij // Nauchnyj servis v seti Internet: trudy XXVI Vserossijskoj nauchnoj konferencii. M.: IPM im. M.V. Keldysha, 2024. S. 39–49.

9. *Paszkowska M. et al.* 3D technologies for Intangible Cultural Heritage Preservation—Literature Review for selected databases. Heritage Science. 2022. Vol. 10(1). No. 3.
URL: https://link.springer.com/article/10.1186/s40494-021-00633-x

10. *Kirillov S.A., Sobolevskaya I.N., Sotnikov A.N.* Principy formirovaniya i predstavleniya mezhdisciplinarnykh kollekcij v cifrovom prostranstve nauchnykh znanij // Russian Digital Library Journal. 2021. T. 24, № 2. S. 294–314.

11. *Botsch M., Pauly M., Kobbelt L., Alliez P., Lévy B., Bischoff S., Röoss C.* Geometric modeling based on polygonal meshes Video files associated with this course are available from the citation page // SIGGRAPH '07: ACM SIGGRAPH 2007 courses. P. 49. https://doi.org/10.1145/1281500.1281640

12. *Russo M.* Polygonal modeling: basic and advanced techniques. Jones &

Bartlett Learning. 2006. 411 p.

13. *Ju T.* Fixing geometric errors on polygonal models: a survey // Journal of Computer Science and Technology. 2009. Vol. 24. No. 1. P. 19–29.

14. *Chuvikov D.A. et al.* 3D modeling and 3D objects creation technology analysis for various intelligent systems // International Journal of Advanced Studies. 2014. Vol. 4. No. 4. P. 16–22.

15. *Beraldin J.A. et al.* Real world modelling through high resolution digital 3D imaging of objects and structures // ISPRS Journal of Photogrammetry and Remote Sensing. 2000. Vol. 55. No. 4. P. 230–250.

16. *Palka D., Sobota M., Buchwald P.* 3D object digitization devices in manufacturing engineering applications and services // Multidisciplinary Aspects of Production Engineering. 2020. Vol. 3. P. 450–463.

17. *Yan Y., Letscher D., Ju T.* Voxel cores: Efficient, robust, and provably good approximation of 3d medial axes // ACM Transactions on Graphics (TOG). 2018. Vol. 37. No. 4. P. 1–13.

18. *Koeva M.N.* 3D modelling and interactive web-based visualization of cultural heritage objects // The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2016. Vol. 41. P. 297–303.

19. Zhengren Wang 3D Representation Methods: A Survey URL: https://arxiv.org/html/2410.06475v1

20. Spatial Representation URL: https://app.uxcel.com/courses/3d-design-foundations/d-spatial-representation-514

21. *Chen C., Chen Z., Zhang J., Tao D*. SASA: Semantics-Augmented Set Abstraction for Point-Based 3D Object Detection // Proceedings of the AAAI Conference on Artificial Intelligence. 2022. Vol. 36, No. 1. P. 221–229.
https://doi.org/10.1609/aaai.v36i1.1989

22. *Wang W. et al.* Volsplat: Rethinking feed-forward 3d gaussian splatting with voxel-aligned prediction // arXiv preprint arXiv:2509.19297. 2025.

23. *Cao R. et al.* Poxel: Voxel Reconstruction for 3D Printing // arXiv preprint arXiv:2501.10474. 2025.

24. What is the Difference Between 3D Point Clouds vs. 3D BIM Models? URL: https://www.existingconditions.com/knowledge-center-articles/3d-point-cloud-vs-model

25. *Yu Y. et al.* From comparison to integration: A workflow evaluation of 3D Gaussian splatting and LiDAR point cloud for modern architectural heritage // Automation in Construction. 2025. Vol. 180. No. 106509.

26. *Kwon O., Yu J.* Realistic and Interactive Virtual Museum Representation Using 3D Gaussian Splatting // ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2025. P. 185–192.

27. *Chibane J., Alldieck T., Pons-Moll G.* Implicit functions in feature space for 3d shape reconstruction and completion // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020. P. 6970–6981.

28. *Bhatnagar B.L. et al.* Combining implicit function learning and parametric models for 3d human reconstruction // European conference on computer vision. Cham: Springer International Publishing, 2020. P. 311–329.

29. *Bison G. et al.* Automatizing 3D reconstruction pipelines for speeding-up cultural heritage digitization // Multimedia Tools and Applications. 2025. P. 1–21.

30. *Kilis N. et al.* AI tools for generating Digital Heritage Twins enhancing storytelling in educational games // Digital Applications in Archaeology and Cultural Heritage. 2025. e00451.

31. *Vargün Ö.* Visual analysis of 3D characters and animations used in the presentation of cultural heritage at museums // Journal of Arts. 2023. Vol. 6. No. 2. P. 141–155.

32. *Sobolevskaya I.N., Sotnikov A.N.* Virtual Exhibition as a Means of Integrating into a Unified Digital Space of Scientific Knowledge the Information Systems in the Field of Science and Culture // Automatic Documentation and Mathematical Linguistics, 2025. Vol. 58. S43–S50. https://doi.org/10.3103/S0005105525700098

33. *Kalenov N.E., Pogorelko K.P., Sobolevskaya I.N., Sotnikov A.N.* Ehlektronnaya biblioteka «Nauchnoe nasledie RossiI» kak ehlement Edinogo cifrovogo prostranstva nauchnykh znanij // Nauchnye i tekhnicheskie bibioteki. 2025. № 8. S. 101–123. https://doi.org/10.33186/1027-3689-2025-8-101-123

34. *Skublewska-Paszkowska M. et al.* 3D technologies for Intangible Cultural Heritage Preservation—Literature Review for selected databases. Heritage Science. 2022. Vol. 10(1). No. 3.
URL: https://link.springer.com/article/10.1186/s40494-021-00633-x.

35. Kruglyj stol «Kul'turnaya vakcina. Muzej v cifrovoj real'nosti» na Peterburgskom mezhdunarodnom yuridicheskom forume – 2021 URL: https://www.hermitagemuseum.org/news/news_107_21?lng=el

36. *Yurina Yu.G., Shalyganova O.S.* KAMIS-2000 v muzee V.G. Belinskogo // V sbornike: Sovremennoe obshchestvo, obrazovanie i nauka. sbornik nauchnykh trudov po materialam Mezhdunarodnoj nauchno-prakticheskoj konferencii: v 16 chastyakh. 2015. S. 143–145.

37. *Rumyancev M.S.* Ispol'zovanie informacionnykh tekhnologij v nauchno-fondovoj deyatel'nosti muzeya (na primere programmy «KAMIS») // V sbornike: XVI ezhegodnaya nauchnaya sessiya aspirantov i molodykh uchenykh. materialy Vserossijskoj nauchnoj konferencii. Vologda, 2023. S. 303–305.

38. *Vlasova S.A., Kalenov N.E.* Dialogovyj programmnyj kompleks formirovaniya ontologii Edinogo cifrovogo prostranstva nauchnykh znanij // Programmnye produkty i sistemy. 2024. T. 37, № 4. S. 514–523.

39. *Vlasova S.A., Kalenov N.E., Kirillov S.A., Sobolevskaya I.N., Sotnikov A.N.* Modelirovanie fragmenta Edinogo cifrovogo prostranstva nauchnykh znanij na primere muzejnykh kollekcij // Nauchno-tekhnicheskaya informaciya. Seriya 1, 2025. № 5. S. 30–35. https://doi.org/10.36535/0548-0019-2025-05-4

40. Gribnoye prostranstvo URL: http://exibitions.jscc.ru/360/15619/

СВЕДЕНИЯ ОБ АВТОРАХ

КИРИЛЛОВ Сергей Александрович – научный сотрудник отделения суперкомпьютерных систем и параллельных вычислений Национального исследовательского центра «Курчатовский институт».

Sergey Aleksandrovich KIRILLOV – researcher at the Department of Supercomputer Systems and Parallel Computing, National Research Center "Kurchatov Institute”**.**

email: skirillov@jscc.ru

ORCID: 0000-0001-7560-0041

СОБОЛЕВСКАЯ Ирина Николаевна– старший научный сотрудник отделения суперкомпьютерных систем и параллельных вычислений Национального исследовательского центра «Курчатовский институт», кандидат физико-математических наук.

Irina Nikolaevna SOBOLEVSKAYA – senior researcher at the Department of Supercomputer Systems and Parallel Computing, National Research Center "Kurchatov Institute”, Candidate of Physics and Math Sciences.

email: nik_first@mail.ru

ORCID: 0000-0002-9461-3750

Материал поступил в редакцию 17 февраля 2026 года

УДК 004.81

ГЕНЕРАЦИЯ ВРЕМЕННЫХ СИГНАЛОВ ИЗ СТАТИЧЕСКИХ ИЗОБРАЖЕНИЙ ДЛЯ ПОДАЧИ НА СПАЙКОВЫЕ НЕЙРОННЫЕ СЕТИ

А. С. Тощев[0000-0003-4424-6822]

Казанский (Приволжский) федеральный университет, г. Казань, Россия

atoschev@kpfu.ru

Аннотация

Спайковые нейронные сети (далее — СНС, т. е. нейросети, передающие информацию во времени с помощью импульсов) требуют временного входа, тогда как в задачах компьютерного зрения данные чаще заданы статическими изображениями. В работе рассмотрено преобразование вида «изображение – временной сигнал – импульсы» и исследовано влияние способа входного кодирования на динамику обучения СНС, плотность импульсной активности и вычислительную стоимость обработки. В экспериментальной части реализованы и сопоставлены два семейства кодирования: кодирование по времени первого импульса (Latency) и пуассоновское кодирование по интенсивности (Poisson); для них рассмотрены четыре режима: базовый Latency без подавления фона, модифицированный Latency с порогом тишины, стохастический Poisson и детерминированный Poisson. В качестве метрик использованы среднее число импульсов на пример, число синаптических операций, прокси-показатель энергозатрат и характеристики конкуренции нейронов скрытого слоя. Эксперименты на наборе MNIST (60000 обучающих и 10000 тестовых изображений) для сети со скрытым слоем из 100 нейронов и горизонтом моделирования 200 шагов показали, что все исследованные режимы обеспечивают устойчивое обучение без коллапса активности. При этом модифицированный Latency с порогом тишины $x_{\min} = 0.05$ оказался наиболее эффективным по соотношению «полезная активность — вычислительная стоимость»: при количестве спайков на один пример 323.41 для него число синаптических операций составило 14295.09, тогда как базовый Latency без фильтрации фона при близкой выходной активности (311.22 импульса на пример) потребовал 78400 синаптических операций.

Ключевые слова: *спайковые нейронные сети, распознавание изображений, кодирование сигнала, кодирование изображений.*

ВВЕДЕНИЕ

Спайковые нейронные сети (СНС) представляют собой класс моделей, в которых вычисления и передача информации задаются во времени, а базовой единицей взаимодействия между нейронами выступает импульс — кратковременное дискретное событие. В отличие от традиционных нейронных сетей, работающих с непрерывными активациями, СНС опираются на временную структуру сигналов и редкие события, что рассматривается как одно из принципиальных отличий третьего поколения нейросетевых моделей [1, 2].

Такой принцип организации делает СНС естественно близкими к биологическим нейронным системам и одновременно открывает путь к событийно-управляемым вычислениям, в которых вычислительная стоимость определяется не столько числом арифметических операций, сколько количеством сгенерированных импульсов и длительностью их обработки. Именно поэтому СНС и нейроморфные системы рассматриваются как перспективное направление для энергоэффективных вычислений, особенно в условиях ограниченных ресурсов и на специализированном аппаратном обеспечении [3].

Однако применение СНС в задачах компьютерного зрения связано с принципиальной трудностью. В большинстве практических постановок входные данные представлены статическими изображениями, тогда как сами СНС наиболее естественно работают с временными или событийными потоками. Это означает, что пространственная структура изображения должна быть специально преобразована во временную последовательность импульсов, причем выбор способа такого преобразования существенно влияет как на качество распознавания, так и на задержку, число событий и вычислительную стоимость. В литературе именно схема кодирования входа рассматривается как один из ключевых факторов, определяющих свойства спайковой сети при обучении и распознавании [4, 5].

Постановка задачи, рассмотренной ниже, заключается в преобразовании статического изображения во временную последовательность импульсных со-

бытий, пригодную для обработки спайковой нейронной сетью. Такое преобразование должно одновременно сохранять информацию, существенную для распознавания, обеспечивать управляемую разреженность импульсной активности и допускать контроль задержки принятия решения. От выбора схемы кодирования в этом случае зависит не только информативность представления, но и режим работы сети в целом [5, 6].

Целью настоящего исследования были разработка и анализ воспроизводимой схемы преобразования изображения во временной вход для спайковых нейронных сетей, а также экспериментальное исследование влияния способа кодирования и его параметров на характеристики импульсной активности и вычислительную стоимость на наборе MNIST [7]. Предложен единый формализм «изображение – временной сигнал – импульсы», выполнено сопоставление реализованных схем кодирования входа, проведено количественное сравнение режимов входного кодирования по показателям импульсной активности, вычислительной стоимости и конкуренции нейронов скрытого слоя и сформирован воспроизводимый набор метрик для анализа таких систем с точки зрения активности, вычислительной стоимости и распределения конкуренции между нейронами [4, 6].

МЕТОДОЛОГИЯ

Были использованы нейроны типа «интегратор с утечкой и порогом» (распространенная модель спайкового нейрона) [8], а обучение в СНС выполнено по правилу пластичности, зависящей от относительного времени импульсов [9].

Требуется построить преобразование статического изображения в последовательность импульсов $X \rightarrow s(t) \rightarrow z(t)$, после чего подать $z(t)$ на СНС и решить задачу классификации. Нас интересует компромисс: точность при ограничениях на число импульсов и задержку [6].

Преобразование «изображение – временной сигнал»

Рассмотрены два семейства кодирования:

1) кодирование по интенсивности (частотное, Poisson);
2) кодирование по времени первого импульса (Latency) [4].

Кодирование по интенсивности (частотное, Poisson)

Пуассоновское кодирование относится к частотным способам представления входа: интенсивность пикселя интерпретируется как интенсивность генерации импульсов во времени. Такой подход широко используется в спайковых сетях для задач распознавания изображений, в том числе в классической постановке Diehl–Cook [4] для MNIST, где входные изображения подаются в виде пуассоновских спайковых последовательностях с частотой, пропорциональной яркости пикселя.

Пусть исходное изображение задано матрицей $X = \{x_{ij}\}$, где $x_{ij} \in [0,1]$. В нашей реализации после нормировки изображение преобразуется в вектор яркостей, а для каждого пикселя задается параметр интенсивности

$$\lambda_{ij} = x_{ij}\beta, \tag{1}$$

где $\beta > 0$ — коэффициент масштабирования интенсивности (rate_scale). Формула (1) задает рабочую точку энкодера: при малых β вход становится слишком разреженным, а при больших β резко растет плотность событий.

Далее вероятность появления импульса на одном шаге времени вычисляется как

$$p_{ij} = 1 - \exp(-\lambda_{ij}). \tag{2}$$

Формула (2) соответствует стандартной связи между интенсивностью пуассоновского процесса и вероятностью события на дискретном временном шаге. В практических реализациях СНС такой способ кодирования используется как базовый вариант частотного представления входа [4].

После этого для каждого шага времени $t = 1, \dots, T$ и каждого пикселя выполняется независимое испытание Бернулли:

$$z_{ij}(t) = \mathbb{I}\big(u_{ij}(t) < p_{ij}\big), \qquad u_{ij}(t) \sim U(0,1), \tag{3}$$

где $z_{ij}(t) \in \{0,1\}$ — наличие импульса на шаге t. В результате получается бинарный импульсный поток длины T, который подается на вход спайковой сети.

Интерпретация параметра β

Параметр β в формулах (1) и (2) управляет компромиссом «информативность – число импульсов». При слишком малых значениях β вероятность p_{ij} близка к нулю, и сеть недополучает событий. При увеличении β растет число входных импульсов и, как правило, повышается информативность входа, однако одновременно возрастают вычислительная стоимость и риск насыщения активности. Сравнительные исследования схем кодирования в СНС также показывают, что частотное кодирование чувствительно к длительности окна и плотности событий, поскольку информация здесь извлекается из накопления импульсов во времени.

Воспроизводимый режим

Поскольку стандартное пуассоновское кодирование стохастично, повторное кодирование одного и того же изображения может давать разные импульсные последовательности. Это допустимо с точки зрения модели, но затрудняет сравнение энкодеров и построение воспроизводимых графиков. Поэтому в реализации предусмотрен детерминированный режим (deterministic = True), в котором псевдослучайная последовательность зависит от изображения и базового зерна генератора. Формально это можно записать в форме

$$u_{ij}(t) = f\,(X, \mathrm{seed}, t, i, j), \tag{4}$$

где функция $f(\cdot)$ строит воспроизводимую псевдослучайную последовательность в интервале $[0,1)$. При этом правило сравнения (3) сохраняется, но одинаковое изображение кодируется одинаково при каждом повторном запуске.

Кодирование по времени первого импульса (Latency)

В этом семействе яркость кодируется моментом появления первого события: яркие пиксели должны давать импульс раньше [10]. Для каждого пикселя вычисляется время

$$t_{ij}^{*} = 1 + \left\lfloor (T-1)(1-x_{ij})^{\gamma} \right\rfloor, \quad \gamma > 0. \tag{5}$$

Далее считаем, что в момент t_{ij}^{*} возникает событие (либо формируется узкий всплеск сигнала вокруг этого момента).

Формула (5) обеспечивает упорядочивание по важности: пиксели с большей яркостью дают меньшие t_{ij}^*, т. е. информация поступает в сеть раньше. Параметр γ в (5) задает «контраст по времени»: при увеличении γ различия между яркими областями сильнее сжимаются к ранним шагам.

Кодирование по времени первого импульса задает для каждого пикселя ровно одно событие, а момент события определяется яркостью: чем ярче пиксель, тем раньше он «срабатывает». В исходной форме такой подход порождает событие для каждого пикселя, включая фон. Для изображений MNIST это приводит к нежелательному эффекту: фоновые пиксели с $x_{ij} \approx 0$ формируют массовый «залп» на последнем шаге времени $t = T - 1$, поскольку для них значение t_{ij}^* оказывается максимальным. Этот залп не несет полезной информации, но резко увеличивает плотность событий и может ухудшать стабильность обучения.

Чтобы устранить данный эффект, в кодировщик был введен порог тишины $x_{\min}$. Пиксели с яркостью ниже $x_{\min}$ считаются фоновыми и не генерируют событие вообще. Таким образом, события формируются только по «значимым» пикселям изображения.

Формально, после нормировки яркости $x_{ij} \in [0,1]$, введем маску активности

$$m_{ij} = \mathbb{I}\left[x_{ij} \geq x_{\min} \right].$$

Для пикселей с $m_{ij} = 1$ вычислим момент первого события:

$$t_{ij}^* = \left\lfloor (1 - x_{ij})(T - 1) \right\rfloor \qquad \text{только для } m_{ij} = 1.$$

Далее импульсный поток зададим в виде

$$z_{ij}(t) = \begin{cases} 1, t = t_{ij}^* \text{ и } m_{ij} = 1, \\ 0, \text{иначе} \end{cases}$$

В результате модификации достигаются два практических эффекта.

1. Устраняется «залп фона» на последнем шаге времени: если $x_{ij} < x_{\min}$, то событие не создается, и фон не формирует массовую активность в момент $t = T - 1$.

2. Снижается число входных событий и повышается информативность временной структуры: события концентрируются на пикселях штрихов цифры, а не на пустом фоне.

В проведенных оценках корректности кодировщика было показано, что для полностью нулевого изображения при $x_{\min} > 0$ поток событий отсутствует, а для разреженных изображений число событий равно числу пикселей, удовлетворяющих условию $x_{ij} \geq x_{\min}$. Это обеспечивает воспроизводимое и интерпретируемое кодирование для дальнейшего обучения и сравнения режимов по метрикам: среднее число импульсов, индекс концентрации победителей, количество задействованных нейронов.

Постановка экспериментов

Было выполнено сравнение четырех режимов кодирования входного изображения в импульсную последовательность:

1) **Latency, all pixels** — базовое кодирование по времени первого импульса без подавления фона (latency_x_min = 0);

2) **Latency, filtered** — модифицированное кодирование по времени первого импульса с порогом тишины (latency_x_min = 0.05);

3) **Poisson, stochastic** — стохастическое пуассоновское кодирование по интенсивности (poisson_deterministic = False);

4) **Poisson, deterministic** — детерминированный пуассоновский режим (poisson_deterministic = True).

Во всех случаях была использована одна и та же архитектура сети: входной слой размерности 784, скрытый слой из 100 нейронов типа «интегратор с утечкой и порогом», горизонт моделирования $T = 200$ шагов. Обучение выполнялось на 60000 изображений MNIST с одинаковыми базовыми гиперпараметрами STDP, инициализации весов, торможения и гомеостатической подстройки порогов. Таким образом, различие между экспериментами определялось только с помощью кодирования входного изображения. Для пуассоновских режимов были использованы poisson_rate_scale = 0.011 и дополнительное усиление входа encoder_rate_boost = 7. Для латентностного режима с фильтрацией фона применялся порог тишины latency_x_min = 0.05, при котором пиксели с яркостью ниже этого значения не генерировали импульсов.

Оцениваемые характеристики

Для каждого режима кодирования анализировалась реакция сети в процессе обучения. Для этого фиксировались следующие метрики:

- среднее число импульсов на пример (spikes_per_sample);
- число синаптических операций на пример (synops_per_sample);
- интегральный прокси-показатель вычислительной стоимости (energy_proxy_per_sample);
- число нейронов, выступавших победителями хотя бы один раз (winners_unique);
- индекс концентрации победителей (winner_HHI).

Такой набор метрик позволяет оценивать не только способность сети обучаться, но и характер формирующейся специализации скрытого слоя: равномерность использования нейронов, степень конкуренции и вычислительную «стоимость» работы сети.

РЕЗУЛЬТАТЫ

Итоги обучения представлены в табл. 1.

Табл. 1. Результаты обучения сети при разных режимах кодирования входа

Режим кодирования	Spikes per sample	Synops per sample	Energy proxy per sample	Winners unique	Winner HHI
Latency, all pixels	311.22	78400.00	4331.22	100	0.01133
Latency, filtered	323.41	14295.09	1138.16	100	0.01119
Poisson, stochastic	451.31	22423.99	1672.51	100	0.01022
Poisson, deterministic	450.36	22426.79	1671.70	100	0.01022

Полученные данные показывают, что все исследованные режимы обеспечивают устойчивое обучение без коллапса активности: во всех случаях winners_unique = 100, т. е. весь скрытый слой участвует в работе, а значения winner_HHI близки к 1/100, что соответствует почти равномерному распределению победителей по нейронам.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Наиболее показательное различие наблюдается между двумя вариантами Latency-кодирования. Базовый режим Latency, all pixels генерирует событие для каждого пикселя, включая фон. Это приводит к крайне большому числу синаптических операций (78400.0 на пример) и высокому энергетическому прокси (4331.22), хотя полезная выходная активность скрытого слоя остается умеренной (311.22 импульса на пример).

Введение порога тишины в режиме Latency, filtered радикально изменяет ситуацию. При latency_x_min = 0.05 фоновая активность подавляется, что приводит к резкому снижению вычислительной стоимости: synops_per_sample уменьшается более чем в 5 раз, а energy_proxy_per_sample — почти в 4 раза. При этом среднее число импульсов скрытого слоя не только не падает, но даже немного увеличивается (323.41 против 311.22). Это означает, что фильтрация фона практически не ухудшает полезную активность сети, но устраняет значительное число нефункциональных входных событий [4, 10].

Оба пуассоновских режима — стохастический и детерминированный — формируют более плотную активность скрытого слоя (≈450 импульсов на пример), чем модифицированный Latency, и имеют более высокую вычислительную стоимость, но остаются существенно более экономичными, чем Latency без фильтрации фона. При этом различия между стохастическим и детерминированным пуассоновским кодированием практически отсутствуют: значения spikes_per_sample, synops_per_sample, energy_proxy_per_sample, winners_unique и winner_HHI совпадают с высокой точностью. Следовательно, детерминированный пуассоновский режим может рассматриваться как воспроизводимая альтернатива стохастическому без заметного изменения характера обучения.

ОГРАНИЧЕНИЯ И БУДУЩАЯ РАБОТА

Несмотря на полученные устойчивые режимы обучения для различных способов кодирования входа, проведенное исследование имеет ряд ограничений.

Во-первых, сравнение выполнялось на относительно простой задаче распознавания рукописных цифр MNIST. Хотя этот набор данных удобен для анализа динамики обучения и свойств кодировщиков, он не отражает всей сложности реальных зрительных задач, где присутствуют текстуры, сложный фон, межклассовое сходство и более высокое пространственное разрешение. Поэтому сделанные выводы о преимуществах того или иного энкодера не следует автоматически переносить на более сложные наборы данных без дополнительной проверки.

Во-вторых, в работе исследована одна фиксированная архитектура сети: входной слой размерности 784, скрытый слой из 100 нейронов типа «интегратор с утечкой и порогом», один обучаемый полносвязный слой и заданный набор параметров STDP, торможения и гомеостатической подстройки порогов. Полученные результаты показали, что даже в такой конфигурации выбор кодировщика существенно влияет на плотность активности и вычислительную стоимость, однако влияние этих факторов может изменяться при увеличении числа нейронов, глубины сети или изменении структуры связей.

В-третьих, оценка на данном этапе сосредоточена преимущественно на динамике обучения сети, а не на окончательном качестве распознавания. В частности, основной акцент сделан на таких метриках, как число импульсов на пример, число синаптических операций, прокси-энергии и распределение победителей по нейронам. Эти метрики важны для понимания устойчивости и вычислительной эффективности сети, однако они не заменяют полноценного анализа итоговой точности классификации после этапа построения карты меток нейронов и последующего выходного классификатора.

В-четвертых, латентностный энкодер был модифицирован введением порога тишины $x_{\min}$, что позволило подавить нефункциональную активность фона. Хотя экспериментально показано, что такая модификация существенно снижает вычислительную стоимость без потери полезной активности, оптимальное значение $x_{\min}$ в настоящей работе систематически не подбиралось. Аналогично,

для пуассоновского кодирования использовалось фиксированное усиление входа, тогда как более подробное исследование по сетке параметров могло бы дать более точные рекомендации по выбору рабочей точки.

Наконец, детерминированный пуассоновский режим рассматривался как воспроизводимая альтернатива стохастическому, однако не анализировалось, в какой степени его использование влияет на итоговую способность сети к обобщению по сравнению с полностью стохастическим режимом.

Направления будущей работы

Перспективы дальнейших исследований связаны как с расширением экспериментальной базы, так и с углублением анализа самих механизмов кодирования.

Во-первых, следующим шагом является полноценная оценка качества классификации для всех исследованных режимов кодирования после построения карты меток нейронов и применения выходных схем распознавания. Это позволит сопоставить не только вычислительную стоимость и структуру активности, но и итоговую точность классификации, а также компромисс «точность — число импульсов».

Во-вторых, представляет интерес систематический перебор параметров энкодеров. Для латентностного кодирования это прежде всего порог тишины $x_{\min}$, а также возможные нелинейные преобразования яркости во время первого импульса. Для пуассоновского кодирования важнейшими параметрами являются коэффициент масштабирования интенсивности и дополнительное усиление входа. Такой анализ позволит построить кривые зависимости качества и вычислительной стоимости от параметров кодирования и выбрать оптимальные рабочие режимы.

В-третьих, перспективно распространить предложенный подход на более сложные наборы данных и архитектуры, включая многослойные спайковые сети, сверточные структуры и более крупные скрытые слои. Это позволит проверить, сохраняется ли наблюдаемое преимущество модифицированного Latency-энкодера по вычислительной эффективности при усложнении задачи.

В-четвертых, важным направлением является исследование более сложных способов входного кодирования, которые в настоящей работе только обсуждались концептуально, но не были реализованы в экспериментальном протоколе. К ним относятся кодирование по изменениям, сканирующие режимы подачи изображения и гибридные схемы, сочетающие ранговую и частотную информацию.

В-пятых, представляет интерес дальнейшее развитие этапа обратного отображения активности сети в метки классов, включая сравнение различных способов построения карты меток нейронов, более сложные схемы голосования по активности скрытого слоя и использование признаков на основе счетчиков импульсов в сочетании с внешними классификаторами.

Таким образом, будущая работа должна объединить три направления: более широкий класс энкодеров, более глубокий анализ параметров и полную оценку итогового качества распознавания, что позволит перейти от анализа динамики обучения к построению практически применимого протокола выбора кодировщика для спайковых сетей.

ЗАКЛЮЧЕНИЕ

Исследовано влияние способа кодирования статического изображения в импульсную последовательность на динамику обучения спайковой нейронной сети. Рассмотрены два основных семейства кодирования, реализованные в экспериментальном протоколе: кодирование по времени первого импульса (Latency) и пуассоновское кодирование по интенсивности (Poisson) в стохастическом и детерминированном режимах. Показано, что при фиксированной архитектуре сети и одинаковых базовых параметрах обучения выбор кодировщика существенно влияет не только на плотность активности скрытого слоя, но и на вычислительную стоимость обработки, выраженную через число синаптических операций и интегральный энергетический прокси.

Ключевым результатом работы стало введение модифицированного латентностного кодировщика с порогом тишины $x_{\min} = 0$, позволяющего подавлять фоновые пиксели и устранять нефункциональный «залп» событий на последнем шаге времени. Экспериментально показано, что такая модификация практиче-

ски не ухудшает полезную выходную активность сети, но резко снижает вычислительную стоимость по сравнению с базовым вариантом Latency без фильтрации фона. В сравнении с пуассоновскими режимами модифицированный Latency продемонстрировал наиболее выгодное соотношение между информативной активностью и стоимостью вычислений, тогда как стохастический и детерминированный Poisson обеспечили режим работы, более плотный, но и более затратный по событиям. При этом детерминированный пуассоновский режим показал практически те же свойства, что и стохастический, оставаясь удобным воспроизводимым вариантом для систематических сравнений.

Таким образом, полученные результаты подтверждают, что выбор энкодера является не второстепенной технической деталью, а одним из центральных факторов, определяющих режим обучения спайковой сети. Предложенный подход позволяет рассматривать кодирование входа как самостоятельный объект оптимизации и сравнивать различные режимы по метрикам «активность – стоимость – специализация нейронов». Это создает основу для дальнейшего перехода от анализа динамики обучения к построению полного протокола распознавания, включающего карту меток нейронов, выходной классификатор и итоговую оценку точности.

Доступность данных и кода

Использован открытый набор данных MNIST [7]. Код экспериментов реализован в виде вычислительного colab-ноутбука и доступен по ссылке https://github.com/alexander-toschev/science-reports/tree/main/energy_efficient.

Благодарности

Работа выполнена за счет гранта, предоставленного Академией наук Республики Татарстан образовательным организациям высшего образования, научным и иным организациям на поддержку планов развития кадрового потенциала в части стимулирования их научных и научно-педагогических работников к защите докторских диссертаций и выполнению научно-исследовательских работ (Соглашение от 22.12.2025 № 12/2025-ПД-КФУ)

Конфликт интересов

Автор заявляет об отсутствии конфликта интересов.

СПИСОК ЛИТЕРАТУРЫ

1. *Gerstner W., Kistler W.* Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge: Cambridge University Press, 2002. https://doi.org/10.1017/CBO9780511815706

2. *Maass W.* Networks of Spiking Neurons: The Third Generation of Neural Network Models // Neural Networks. 1997. Vol. 10, No. 9. P. 1659–1671. https://doi.org/10.1016/S0893-6080(97)00011-7

3. *Davies M. et al.* Loihi: A Neuromorphic Manycore Processor with On-Chip Learning // IEEE Micro. 2018. Vol. 38, No. 1. P. 82–99. https://doi.org/10.1109/MM.2018.112130359

4. *Diehl P.U., Cook M.* Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity // Frontiers in Computational Neuroscience. 2015. Vol. 9. Art. 99. https://doi.org/10.3389/fncom.2015.00099

5. *Rueckauer B. et al.* Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification // Frontiers in Neuroscience. 2017. Vol. 11. Art. 682. https://doi.org/10.3389/fnins.2017.00682

6. *Guo W. et al.* Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems // Frontiers in Neuroscience. 2021. Vol. 15. Art. 638474. https://doi.org/10.3389/fnins.2021.638474

7. *LeCun Y., Bottou L., Bengio Y., Haffner P.* Gradient-Based Learning Applied to Document Recognition // Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324. https://doi.org/10.1109/5.726791

8. *Izhikevich E.M.* Which Model to Use for Cortical Spiking Neurons? // IEEE Transactions on Neural Networks. 2004. Vol. 15, No. 5. P. 1063–1070. https://doi.org/10.1109/TNN.2004.832719

9. *Caporale N., Dan Y.* Spike Timing-Dependent Plasticity: A Hebbian Learning Rule // Annual Review of Neuroscience. 2008. Vol. 31. P. 25–46. https://doi.org/10.1146/annurev.neuro.31.060407.125639

10. *Thorpe S., Delorme A., Van Rullen R.* Spike-Based Strategies for Rapid Processing // Neural Networks. 2001. Vol. 14, No. 6–7. P. 715–725. https://doi.org/10.1016/S0893-6080(01)00083-1

11. *Hazan H. et al.* BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python // Frontiers in Neuroinformatics. 2018. Vol. 12. Art. 89. https://doi.org/10.3389/fninf.2018.00089

GENERATING TEMPORAL SIGNALS FROM STATIC IMAGES FOR SPIKING NEURAL NETWORKS

A. S. Toshchev[0000-0003-4424-6822]

Kazan Federal University, Kazan, Russia

atoschev@kpfu.ru

Abstract

Spiking neural networks (SNNs), i.e., neural architectures that represent and transmit information in the form of temporally distributed spikes, require time-dependent input, whereas data in computer vision are most commonly available as static images. This study addresses the transformation pipeline “image → temporal signal → spikes” and examines how the choice of input encoding influences SNN training dynamics, spike activity density, and computational cost. The experimental section implements and compares two encoding families: first-spike-time encoding (Latency) and intensity-based Poisson encoding (Poisson). Within these families, four operating modes are considered: baseline Latency without background suppression, modified Latency with a silence threshold, stochastic Poisson, and deterministic Poisson. The evaluation employs the following metrics: the average number of spikes per sample, the number of synaptic operations, an energy-related proxy metric, and indicators characterizing competition among hidden-layer neurons. Experiments conducted on the MNIST dataset (60000 training and 10000 test images) using a network with a hidden layer of 100 neurons and a simulation horizon of 200 time steps demonstrate that all examined modes support stable learning without activity collapse. Among them, the modified Latency mode with a silence threshold of $x_{\min}$ = 0.05 achieves the most favorable balance between useful activity and computational cost: at 323.41 spikes per sample, it requires 14925.09 synaptic operations, whereas the baseline Latency mode without background filtering, despite exhibiting a comparable

level of output activity (311.22 spikes per sample), requires 78400 synaptic operations.

Keywords: *spiking neural networks, image recognition, signal encoding, image encoding.*

REFERENCES

1. *Gerstner W., Kistler W.* Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge: Cambridge University Press, 2002. https://doi.org/10.1017/CBO9780511815706

2. *Maass W.* Networks of Spiking Neurons: The Third Generation of Neural Network Models // Neural Networks. 1997. Vol. 10, No. 9. P. 1659–1671. https://doi.org/10.1016/S0893-6080(97)00011-7

3. *Davies M. et al.* Loihi: A Neuromorphic Manycore Processor with On-Chip Learning // IEEE Micro. 2018. Vol. 38, No. 1. P. 82–99. https://doi.org/10.1109/MM.2018.112130359

4. *Diehl P.U., Cook M.* Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity // Frontiers in Computational Neuroscience. 2015. Vol. 9. Art. 99. https://doi.org/10.3389/fncom.2015.00099

5. *Rueckauer B. et al.* Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification // Frontiers in Neuroscience. 2017. Vol. 11. Art. 682. https://doi.org/10.3389/fnins.2017.00682

6. *Guo W. et al.* Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems // Frontiers in Neuroscience. 2021. Vol. 15. Art. 638474. https://doi.org/10.3389/fnins.2021.638474

7. *LeCun Y., Bottou L., Bengio Y., Haffner P.* Gradient-Based Learning Applied to Document Recognition // Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324. https://doi.org/10.1109/5.726791

8. *Izhikevich E.M.* Which Model to Use for Cortical Spiking Neurons? // IEEE Transactions on Neural Networks. 2004. Vol. 15, No. 5. P. 1063–1070. https://doi.org/10.1109/TNN.2004.832719

9. *Caporale N., Dan Y.* Spike Timing-Dependent Plasticity: A Hebbian Learning Rule // Annual Review of Neuroscience. 2008. Vol. 31. P. 25–46. https://doi.org/10.1146/annurev.neuro.31.060407.125639

10. *Thorpe S., Delorme A., Van Rullen R.* Spike-Based Strategies for Rapid Processing // Neural Networks. 2001. Vol. 14, No. 6–7. P. 715–725. https://doi.org/10.1016/S0893-6080(01)00083-1

11. *Hazan H. et al.* BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python // Frontiers in Neuroinformatics. 2018. Vol. 12. Art. 89. https://doi.org/10.3389/fninf.2018.00089

СВЕДЕНИЯ ОБ АВТОРЕ

ТОЩЕВ Александр Сергеевич – заведующий кафедрой, канд. техн. наук, доцент, КФУ / Институт информационных технологий и интеллектуальных систем / Кафедра цифровой аналитики и технологий искусственного интеллекта, г. Казань.

Aleksandr Sergeevich TOSHCHEV – Department Head, Ph.D. in Engineering, associate professor at Kazan Federal University (KFU), Institute of Information Technologies and Intelligent Systems, Department of Digital Analytics and Artificial Intelligence Technologies, Kazan.

email: atoschev@kpfu.ru

ORCID: 0000-0003-4424-6822

Материал поступил в редакцию 16 марта 2026 года