

Федеральный исследовательский центр «Информатика и управление»
Российской академии наук

На правах рукописи



Ватолин Алексей Сергеевич

**Обучение и оценивание мультязычных нейросетевых моделей
семантического векторного представления научных текстов**

Специальность 2.3.8 —
«Информатика и информационные процессы»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор физико-математических наук, доцент
Абгарян Каринэ Карленовна

Москва — 2026

Оглавление

| | Стр. |
|--|-----------|
| Введение | 5 |
| Глава 1. Семантическое векторное представление текстов | 13 |
| 1.1 Задача семантического векторного представления текстов | 14 |
| 1.2 Токенизация текстовых данных | 15 |
| 1.3 Архитектура трансформер-кодировщик | 18 |
| 1.4 Предобучение моделей на основе архитектуры трансформер | 22 |
| 1.5 Контрастивное дообучение для векторных представлений текста | 25 |
| 1.6 Применение и оценка моделей в научной области | 30 |
| 1.7 Оценка качества моделей векторного представления текстов | 36 |
| 1.7.1 Бенчмарк SciDocs | 37 |
| 1.7.2 Бенчмарк SciRepEval | 39 |
| 1.7.3 Бенчмарк SciFact | 40 |
| 1.7.4 Универсальные бенчмарки: МТЕВ | 42 |
| 1.8 Основные выводы | 42 |
| Глава 2. Разработка и обучение двуязычных моделей векторизации научных текстов SciRus | 44 |
| 2.1 Постановка задачи и существующие ограничения | 44 |
| 2.2 Наборы данных для обучения | 46 |
| 2.3 Архитектура моделей SciRus | 49 |
| 2.4 Методология обучения | 50 |
| 2.4.1 Предобучение с использованием Masked Language Modeling (MLM) | 50 |
| 2.4.2 Контрастивное дообучение | 53 |
| 2.5 Оценка качества моделей SciRus | 54 |
| 2.6 Оценка производительности | 59 |
| 2.7 Практическая апробация и внедрение модели SciRus-tiny | 60 |
| 2.8 Основные выводы | 62 |

| | |
|--|------------|
| Глава 3. Мультизадачный бенчмарк для оценки моделей векторного представления русско- и англоязычных научных текстов . . . | 64 |
| 3.1 Источники данных и подготовка корпуса | 65 |
| 3.2 Состав и методология оценки в бенчмарке RuSciBench | 66 |
| 3.2.1 Задачи классификации документов | 68 |
| 3.2.2 Задачи регрессии | 75 |
| 3.2.3 Задачи информационного поиска | 79 |
| 3.2.4 Задачи кросс-языкового поиска | 82 |
| 3.3 Оценка моделей на RuSciBench | 85 |
| 3.3.1 Оценка степени языковой специализации моделей | 88 |
| 3.4 Интеграция в международный бенчмарк MTEB | 91 |
| 3.5 Основные выводы | 93 |
| Глава 4. Бенчмарк для оценки качества верификации научных фактов на русском языке | 94 |
| 4.1 Постановка задачи верификации научных фактов | 94 |
| 4.2 Методология формирования набора данных RuSciFact | 96 |
| 4.3 Конвейер генерации и фильтрации данных | 99 |
| 4.3.1 Генерация подтверждающих утверждений | 100 |
| 4.3.2 Генерация противоречащих утверждений | 102 |
| 4.4 Экспертная валидация и характеристики набора данных | 104 |
| 4.5 Экспериментальная оценка и анализ результатов | 105 |
| 4.5.1 Оценка в задаче информационного поиска | 107 |
| 4.5.2 Оценка в задаче классификации | 109 |
| 4.6 Основные выводы | 111 |
| Заключение | 113 |
| Список литературы | 115 |
| Приложение А. Промпты для формирования выборки в бенчмарке RuSciFact и примеры данных | 124 |
| A.1 Промпты для генерации подтверждающего утверждения в бенчмарке RuSciFact | 124 |
| A.2 Промпт для классификации сложности факта | 127 |

| | Стр. |
|---|------|
| A.3 Промпт для генерации опровергающего факта | 129 |
| A.4 Промпт для оценки релевантности и степени подтверждения | 131 |
| A.5 Примеры положительных и отрицательных утверждений из RuSciFact | 132 |

Введение

Актуальность данной работы обусловлена кратным увеличением объема публикуемой научной информации. Исследование динамики наполнения крупных библиометрических баз данных подтверждает эту тенденцию. Так, согласно анализу базы данных Scopus, количество ежегодно индексируемых научных статей выросло с приблизительно 921 тысячи в 2000 году до более чем 2,57 миллиона в 2020 году, что свидетельствует о почти трехкратном увеличении за два десятилетия [1]. Общий объем публикаций в Scopus к 2020 году превысил 56 миллионов единиц. Аналогичные тенденции наблюдаются и в других международных наукометрических системах, таких как Web of Science. Российская научная электронная библиотека eLibrary.ru, которая представляет материалы как на русском, так и на других языках, также демонстрирует значительный рост: количество публикаций в год увеличилось с 45,7 тысяч в 2000 году до более чем 4,76 миллиона в 2020 году [2]. Количество публикаций увеличивается не только на английском языке. Этот информационный поток делает задачу поиска релевантной информации для исследователей всё более трудной, а также ставит новые вызовы по эффективной обработке и анализу постоянно растущих текстовых массивов, что требует применения высокопроизводительных подходов, в том числе методов параллельной обработки данных [3].

Методы анализа и поиска информации развиваются в направлении более глубокого учета смысла и контекста обрабатываемых текстов. Классические статистические модели, включая векторную модель поиска VSM [4] и вероятностную модель BM25 [5], остаются широко используемыми методами информационного поиска, однако в задачах семантического анализа научных текстов все большую роль играют нейросетевые подходы. Наиболее заметное развитие связано с архитектурой трансформер [6], позволяющей строить контекстно-зависимые представления текста, учитывать семантическую близость терминов, обрабатывать синонимию и решать задачи многоязычного, а также кросс-языкового поиска. Использование таких моделей в научно-информационных системах повышает качество поиска и расширяет возможности автоматизированного анализа содержания публикаций [7] [8]. В настоящей диссертации представлены нейросетевые модели SciRus [9], предназначенные для работы с научными публикациями. Одна из разработанных моделей, SciRus-tiny, внедрена на сервисе eLibrary.ru.

Развитие моделей обработки естественного языка требует воспроизводимых способов их сравнения. Для этой цели применяются бенчмарки, то есть специализированные наборы данных и задач, позволяющие оценивать качество моделей в сопоставимых условиях. Для русского языка уже предложены универсальные инструменты такой оценки, в том числе RuSentEval [10] и encodechka [11]. Вместе с тем универсальные наборы данных не отражают в полной мере специфику научного дискурса: высокую информационную плотность, терминологическую насыщенность, устойчивые жанровые и композиционные особенности, а также особую роль библиографических и тематических связей между публикациями. Поэтому результаты, полученные на общезыковых тестовых наборах, не дают достаточного основания для вывода о качестве модели при работе с научными текстами.

Отсутствие специализированных русскоязычных научных наборов данных осложняет объективную оценку и сравнительный анализ как существующих, так и вновь разрабатываемых моделей векторного представления текстов в данной предметной области. Это, в свою очередь, ограничивает развитие методов автоматизированной обработки русскоязычного научного контента и затрудняет практический выбор модели для конкретных исследовательских и прикладных задач.

В связи с этим разработка открытых моделей и инструментов оценки, включая SciRus и RuSciBench, имеет не только методическое, но и прикладное значение для научного сообщества. Указанное направление соответствует целям и задачам развития искусственного интеллекта в Российской Федерации, закрепленным в «Национальной стратегии развития искусственного интеллекта на период до 2030 года» (с изменениями от 15 февраля 2024 г.) [12]. Это направление деятельности полностью соответствует концепции открытой науки, предполагающей свободный доступ к исследовательским данным, инструментам и результатам, что способствует ускорению научного прогресса и повышению прозрачности исследовательской деятельности.

Задача построения векторных представлений текста имеет длительную историю и первоначально решалась методами статистического анализа: векторной моделью поиска, взвешиванием термов, вероятностными моделями ранжирования и латентно-семантическим анализом [4; 5; 13; 14]. Следующий этап связан с нейросетевыми представлениями слов и текстов, в том числе с моделями word2vec, GloVe и ELMo [15—17]. Существенное развитие область получила

после появления архитектуры трансформер [6] и моделей семейства BERT, обучаемых на больших текстовых корпусах с помощью маскированного языкового моделирования [18]. Вместе с тем такие модели изначально не предназначены для построения единого вектора предложения или документа, пригодного для быстрого семантического поиска. Для решения этой проблемы были предложены сиамские архитектуры и методы контрастивного дообучения, в том числе Sentence-BERT и функция потерь InfoNCE [19; 20].

Отдельное направление исследований связано с обработкой научных текстов, поскольку научная публикация отличается от общезыковых текстов терминологической насыщенностью, высокой плотностью смысла, устойчивой структурой и наличием ссылочных связей между работами. Для учета этих особенностей были разработаны специализированные модели и методы: SciBERT использует корпус научных публикаций [21], SPECTER обучается с учетом графа цитирований [22], SciRepEval расширяет оценку представлений научных документов на несколько типов задач [23], а методы контрастивного обучения на соседях в графе цитирований дополнительно используют структуру научных связей между публикациями [24]. Эти работы показали, что учет особенностей научного текста и структуры цитирований повышает качество представления научных документов. Однако основная часть таких исследований выполнена для англоязычных корпусов, а вопрос построения компактных моделей, одновременно пригодных для русско- и англоязычных научных текстов, остается недостаточно изученным.

Существенно развита и область оценки моделей векторного представления текста. Для общей оценки текстовых представлений используются MTEB и BEIR [25; 26]; для русского языка предложены RuSentEval, encodechka и ruMTEB [10; 11; 27]. При этом такие инструменты не позволяют в полной мере оценить качество моделей именно на научных текстах: они не отражают специфику научной терминологии, библиографических связей, тематической классификации публикаций, наукометрических признаков и поиска между русскими и английскими научными документами. Следовательно, до настоящей работы отсутствовал специализированный открытый инструментарий, позволяющий сопоставлять модели векторного представления на русско- и англоязычных научных данных в единой воспроизводимой процедуре оценки.

Смежной задачей является проверка научных утверждений по текстам публикаций. Для английского языка эта задача представлена, в частности, набором

SciFact, в котором проверяется соответствие утверждения научному источнику [28]. В последние годы также исследуется автоматическое и полуавтоматическое формирование обучающих примеров для проверки фактов, включая генерацию утверждений по исходным текстам [29—31]. Для русского языка существуют работы по проверке фактов и выявлению противоречий [32; 33], однако открытые наборы данных, предназначенные именно для проверки научных фактов, ранее не были представлены. Недостаточно изучены и методы формирования таких наборов, совмещающие генерацию утверждений, машинную предварительную проверку и экспертную валидацию. Таким образом, степень разработанности темы характеризуется наличием развитой методической базы для векторизации текста и англоязычной обработки научных публикаций, но также наличием существенных пробелов в области компактных русско-англоязычных моделей, специализированной оценки на научных данных и проверки научных фактов на русском языке.

Целью данной работы является разработка, исследование и апробация инструментов и моделей, предназначенных для решения задач эффективной обработки, анализа и оценки качества представления научных текстов на русском языке.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Разработать методику обучения легковесной двуязычной модели для эффективного векторного представления научных текстов на русском и английском языках. Исследовать подходы к обучению, основанные на доступных данных из мультязычных корпусов, без дополнительной разметки.
2. Разработать методологию и на ее основе создать инструментарий для оценки качества векторных представлений научных текстов на русском и английском языках. Данный инструментарий должен учитывать специфику научного дискурса и охватывать разнообразные задачи, используя данные из российской научной среды.
3. Исследовать проблему верификации научных фактов на русском языке. Разработать и апробировать методологию полуавтоматизированного формирования русскоязычного набора данных, включающую генерацию научных утверждений на основе аннотаций с использованием больших языковых моделей и их последующую экспертную валидацию. Разрабо-

тать тестовый набор на основе данного набора для оценки способности моделей определять соответствие или противоречие утверждений.

Основные положения, выносимые на защиту:

1. Предложены компактные двуязычные модели SciRus-tiny (23 млн параметров) и SciRus-small (61 млн параметров) для представления научных текстов в векторном пространстве. Обучение проводится в два этапа: сначала модель обучается с помощью маскированного языкового моделирования, затем с помощью контрастивного дообучения на парах «заголовков-аннотация». Дополнительно, для формирования более сильного обучающего сигнала, при обучении используются пары «цитирующая статья — цитируемая статья», основанные на однонаправленной связи из графа цитирований. Эксперименты на русскоязычных и англоязычных научных наборах для оценки показали, что при числе параметров, в 24 и 9 раз меньшем по сравнению с multilingual-e5-large-instruct (560 млн параметров), предлагаемые модели демонстрируют сопоставимый, а на русскоязычных задачах — превосходящий уровень качества.
2. Разработан мультизадачный двуязычный бенчмарк RuSciBench, включающий задачи классификации, регрессии, моно- и кросс-языкового поиска на научных данных. Этот тестовый набор обеспечивает воспроизводимую процедуру тестирования, интегрированную в международный лидерборд MTEB.
3. Предложена полуавтоматическая методика формирования наборов данных для проверки научных фактов на русском языке, сочетающая генерацию утверждений с помощью LLM, многоступенчатую самооценку модели и экспертную верификацию. На её основе создан первый русскоязычный набор данных для проверки научных фактов RuSciFact.

Методы исследования. В диссертационном исследовании использованы известные, достоверные и хорошо зарекомендовавшие себя на практике методы. В модели используется архитектура трансформер, она обучается с помощью маскированного языкового моделирования и с помощью контрастивного дообучения с применением функции потерь InfoNCE. В наборах для оценки используются распространенные критерии оценки качества, такие как Accuracy, F1-мера, NDCG@k, MRR@k, коэффициент корреляции Кендалла.

Научная новизна:

1. Показана эффективность контрастивного дообучения модели векторизации текста на парах «заголовок-аннотация» и на парах цитирующая-цитируемая статья.
2. Разработаны легковесные модели векторизации научных текстов.
3. Разработан первый набор для оценки качества работы моделей с научными данными, состоящий из различных типов задач.
4. Впервые предложена полуавтоматизированная многоступенчатая методика формирования наборов данных для проверки научных фактов на русском языке, совмещающая генерацию утверждений с помощью LLM, самокритичную оценку модели-генератора и экспертную валидацию.
5. Разработан и опубликован первый набор данных для проверки научных фактов на русском языке RuSciFact.

Практическая значимость обусловлена разработкой открытых научных наборов для оценки, которые были внедрены в авторитетный международный бенчмарк MTEB (Massive Text Embedding Benchmark), что подтверждает их актуальность, а также существенно упрощает их использование для разработчиков моделей. Кроме того, данные RuSciBench были использованы при формировании одной из задач мультязычного набора для оценки AIRBench[34] и одной из задач русскоязычного тестового набора LIBRA[35]. Это свидетельствует о востребованности полученных результатов в научном сообществе.

Практическая значимость работы также подтверждается внедрением модели векторизации научных текстов SciRus-tiny на российском научном портале eLibrary.ru. На основе этой модели был разработан режим «нейропоиск», предназначенный для поиска тематически близких научных публикаций по аннотации статьи. Такое внедрение расширяет возможности поиска и анализа научной информации для исследователей и специалистов, работающих с научной библиотекой eLibrary.ru.

Теоретическая значимость работы заключается в развитии методических основ обучения и оценивания мультязычных нейросетевых моделей семантического векторного представления научных текстов в условиях доменной и языковой специфики. В работе систематизированы и экспериментально исследованы факторы, влияющие на качество таких моделей в русско- и англоязычном научном домене: использование доменно-специфического корпуса, контрастивного обучения на парах «заголовок–аннотация», обучающего сигнала из графа цитирований,

размера модели и степени ее языковой специализации. На основе разработанного двуязычного мультиязычного бенчмарка количественно оценены различия в качестве работы современных моделей на русском и английском языках, что уточняет границы применимости универсальных мультиязычных трансформерных моделей к специализированным научным текстам. Дополнительный теоретический вклад состоит в предложенной методологии формирования и анализа наборов данных для верификации научных фактов на русском языке, позволяющей разделять способность моделей к семантическому поиску и способность к логическому сопоставлению научного утверждения с текстом аннотации.

Достоверность полученных результатов подтверждается следующим:

1. докладами и обсуждениями результатов на международных конференциях
2. публикациями результатов в рецензируемых научных изданиях, рекомендованных ВАК
3. открытым исходным кодом и воспроизводимостью результатов.

Апробация работы. Основные результаты работы докладывались на:

1. А. С. Ватолин. Сравнительный анализ современных мультиязычных моделей для векторизации текста на русском языке. *Международная научно-практическая конференция «Информационные технологии, искусственный интеллект, большие данные: актуальные тенденции, перспективные исследования»*, 2024
2. А. С. Ватолин. ruSciFact: Open Benchmark for Verifying Scientific Facts in Russian. *Международная конференция по компьютерной лингвистике и интеллектуальным технологиям «Диалог»*, 2025
3. А. С. Ватолин. Structured Sentiment Analysis with Large Language Models: A Winning Solution for RuOpinionNE-2024. *Международная конференция по компьютерной лингвистике и интеллектуальным технологиям «Диалог»*, 2025

Личный вклад соискателя в работах с соавторами заключается в следующем: [9] - предобучение маленькой версии модели (SciRus-tiny) с помощью маскированного языкового моделирования, дообучение обеих версий модели (SciRus-tiny и SciRus-small) на парах заголовков-аннотация, валидация моделей на наборе данных SciDocs. [36] - сбор датасетов для классификации по ГРНТИ, по типу публикации, для поиска цитирований, для регрессии по количеству цитат, для поиска английского перевода по тексту на русском языке. Также реализа-

ция исходного кода инструментария для оценки, валидация моделей, интеграция бенчмарка в международный бенчмарк МТЕВ. [37] - вклад соискателя является определяющим. [38] — в рамках работы над созданием и расширением международного многоязычного бенчмарка ММТЕВ соискателем проведена работа по добавлению новых задач на различных языках, включая задачи, разработанные им самостоятельно. Также выполнен значительный вклад в обеспечение качества и корректности данных во всех задачах, вошедших в итоговый набор бенчмарка. В публикации [39] соискатель является единственным автором.

Содержание диссертации и положения, выносимые на защиту, отражают персональный вклад автора в опубликованных работах. Результаты, составляющие основу диссертационного исследования и выносимые на защиту, получены автором лично либо при его определяющем участии; вклад автора в работах, выполненных в соавторстве, указан выше.

Публикации. Основные результаты по теме диссертации изложены в 5 печатных изданиях: 2 — в рецензируемых научных журналах, индексируемых в Scopus и РИНЦ, 3 — в полнотекстовых статьях в рецензируемых трудах международных конференций, индексируемых в Scopus.

Объем и структура работы. Диссертация состоит из введения, 4 глав, заключения и 1 приложения. Полный объём диссертации составляет 134 страницы, включая 9 рисунков и 23 таблицы. Список литературы содержит 76 наименований.

Глава 1. Семантическое векторное представление текстов

Настоящая глава посвящена систематическому обзору методов построения семантических векторных представлений текстов, являющихся основой для решения широкого спектра задач обработки естественного языка. Вначале формулируется сама задача векторизации, определяется ее цель — построение отображения из пространства текстов в векторное пространство, в котором геометрическая близость векторов отражает семантическую близость исходных текстов, и дается краткий исторический экскурс в развитие подходов к ее решению (раздел 1.1). Далее детально рассматривается технологический стек, лежащий в основе современных моделей. Описывается неотъемлемый этап предобработки — субсловная токенизация данных (раздел 1.2), после чего подробно излагается архитектура трансформер-кодировщика (раздел 1.3), ставшая стандартом в данной области.

Особое внимание уделяется современным парадигмам обучения. Рассматривается этап предобучения на больших неразмеченных корпусах текста с использованием таких задач, как маскированное языковое моделирование, на примере моделей BERT и RoBERTa (раздел 1.4). Далее рассматривается причина, по которой предобученные языковые модели не всегда пригодны для семантического поиска без дополнительной адаптации. В этой связи обосновывается необходимость дообучения (fine-tuning) и описываются сиамские архитектуры, а также контрастивные функции потерь Triplet Loss и InfoNCE, используемые для формирования векторного пространства с выраженной семантической структурой (раздел 1.5).

После обзора общих методов рассматривается их применение к научным текстам и адаптация к особенностям этой предметной области. В разделе анализируются свойства научного дискурса, приводящие к доменному сдвигу, и модели, специально разработанные для его учета, включая SPECTER, SPECTER2 и SciNCL (раздел 1.6). Завершает главу обзор бенчмарков и мер качества, используемых для оценки моделей векторизации научных документов. В их число входят SciDocs, SciRepEval, SciFact и универсальный набор MTEB (раздел 1.7).

1.1 Задача семантического векторного представления текстов

Одной из фундаментальных задач в области обработки естественного языка является векторизация, или получение семантических векторных представлений (эмбеддингов), текстовых данных. Суть этой задачи заключается в построении отображения текстовых единиц, будь то слова, предложения или целые документы, в многомерное векторное пространство \mathbb{R}^d фиксированной размерности. Такое отображение переводит исходные текстовые данные, являющиеся по своей природе дискретными и неструктурированными, в формат непрерывных векторов. Это позволяет формировать метрическое пространство, в котором геометрическая близость векторов соответствует семантической близости текстов. Появляется возможность количественно оценивать семантические отношения через вычисление расстояний и углов между векторами, что является математической основой для решения широкого спектра прикладных задач, включая информационный поиск, машинный перевод, классификацию документов, кластеризацию, ответы на вопросы и многие другие [40].

Формально, задачу векторизации можно поставить следующим образом. Пусть дана коллекция текстовых документов $\mathcal{D} = \{x_i\}_{i=1}^N$, где каждый документ x_i представляет собой последовательность токенов (слов или их частей) $x_i = (w_{i,1}, w_{i,2}, \dots, w_{i,n_i})$. Необходимо найти такое параметризованное отображение $f(x, \alpha)$, которое ставит в соответствие каждому документу x_i вещественный вектор $\mathbf{v}_i \in \mathbb{R}^d$:

$$\mathbf{v}_i = f(x_i, \alpha)$$

где α - вектор обучаемых параметров модели. Основная цель обучения состоит в подборе таких параметров α , при которых векторы \mathbf{v}_i наилучшим образом кодируют семантическую информацию, содержащуюся в исходных текстах x_i . Качество этого кодирования оценивается через успешность решения целевых прикладных задач.

Исторически первыми подходами к решению этой задачи были методы, основанные на дистрибутивной гипотезе, утверждающей, что слова, встречающиеся в схожих контекстах, имеют близкие значения [41]. Это привело к появлению статистических моделей, основанных на взвешивании термов, таких как TF-IDF (Term Frequency-Inverse Document Frequency) [13], и методов,

использующих матричные разложения для выявления скрытых тем, например, латентно-семантического анализа (LSA) [14]. Однако эти подходы не всегда эффективно улавливали тонкие семантические связи.

Прорыв в этой области был связан с появлением плотных векторных представлений (dense embeddings), обучаемых с помощью нейронных сетей [42]. Модели семейства word2vec [15] и GloVe [16] позволили получать векторные представления для отдельных слов. Однако они присваивали каждому слову единственный вектор, не учитывая многозначность и контекст его употребления. Развитием этой идеи стали модели, способные генерировать контекстуализированные векторные представления. Модель ELMo (Embeddings from Language Models) [17] использовала для этой цели двунаправленную рекуррентную нейронную сеть (LSTM), выходы которой на разных слоях объединялись для получения итогового вектора токена, зависящего от контекста. Несмотря на успех, рекуррентные архитектуры обрабатывают текст последовательно, что затрудняет распараллеливание вычислений и улавливание дальних зависимостей в тексте.

Современный этап развития методов векторизации неразрывно связан с архитектурой трансформер (Transformer) [6], которая полностью отказалась от рекуррентных связей в пользу механизма внимания (attention mechanism). Этот подход продемонстрировал выдающуюся эффективность и масштабируемость, лег в основу большинства передовых моделей обработки естественного языка.

1.2 Токенизация текстовых данных

Первым и обязательным шагом при обработке текста нейросетевыми моделями является **токенизация** — процесс разбиения сплошного текста на последовательность элементарных единиц, называемых токенами. Эти токены затем отображаются в числовые идентификаторы в соответствии с заранее определенным словарем \mathcal{V} , что позволяет представить исходную текстовую последовательность в виде вектора целых чисел, пригодного для подачи на вход нейронной сети.

Простейшим подходом является токенизация по словам, где словарь состоит из всех уникальных слов, встретившихся в обучающем корпусе. Однако этот метод сталкивается с серьезной проблемой «неизвестных слов» (out-of-

vocabulary, OOV), когда в обрабатываемом тексте появляются слова, отсутствовавшие в обучающих данных. Эта проблема особенно остра для морфологически богатых языков, таких как русский, где одно и то же слово может иметь множество форм.

Для решения этой проблемы были разработаны алгоритмы **субсловной токенизации** (subword tokenization), которые разбивают слова на более мелкие, но семантически значимые части. Такой подход позволяет, с одной стороны, сохранить в словаре наиболее частотные слова целиком, а с другой — представить редкие и неизвестные слова как последовательность известных субсловесных единиц. Это не только решает проблему OOV, но и позволяет модели улавливать морфологические связи между словами (например, «обучение» и «обучать» будут иметь общие токены). К наиболее известным алгоритмам субсловной токенизации относятся Byte-Pair Encoding (BPE) [43], WordPiece [44] и Unigram Language Model [45].

Алгоритм Byte-Pair Encoding (BPE) , изначально разработанный для сжатия данных, был успешно адаптирован для задач обработки естественного языка в работе [43]. Процесс его применения состоит из двух этапов: обучения словаря и токенизации нового текста.

Процесс формирования словаря является итеративным.

1. **Инициализация.** Исходный словарь состоит из всех уникальных символов (символьный алфавит), встречающихся в обучающем корпусе. Весь текст корпуса разбивается на последовательности этих символов.
2. **Итеративное слияние.** На каждой итерации в корпусе находится наиболее часто встречающаяся пара соседних токенов (например, пара символов 'т' и 'о').
3. **Обновление.** Эта пара объединяется в новый, единый токен ('то'), который добавляется в словарь. Все вхождения исходной пары в корпусе заменяются на новый токен.
4. **Повторение.** Шаги 2 и 3 повторяются заданное число раз. Это число (количество слияний) является гиперпараметром, который определяет итоговый размер словаря.

В результате получается словарь, состоящий из исходных символов и наиболее частотных субсловесных единиц различной длины.

Для токенизации нового текста выполняются те же операции слияния в том порядке, в котором они были выучены на этапе обучения. Текст сначала разбивается на символы, а затем к нему последовательно применяются правила слияния до тех пор, пока не останется пар, подлежащих объединению.

Токенизация на уровне байтов (Byte-level BPE) Стандартный алгоритм BPE, работающий на уровне символов, все еще может столкнуться с проблемой OOV, если во входном тексте встретится символ, отсутствовавший в обучающих данных (например, редкий иероглиф или эмодзи). Чтобы полностью устранить эту проблему, была предложена модификация **byte-level BPE**, которая используется в таких моделях, как RoBERTa [46] и GPT-2 [47].

Ключевая идея этого подхода состоит в том, чтобы работать не с символами Unicode, а с их байтовым представлением в кодировке UTF-8. Это гарантирует, что любой текст может быть представлен без потерь и без использования специального токена неизвестного слова.

Принцип работы:

1. **Инициализация.** Начальный словарь состоит из всех 256 возможных значений байтов (от 0x00 до 0xFF). Таким образом, любой текст, представленный в виде последовательности байтов, может быть изначально токенизирован без потерь.
2. **Обучение.** Процесс обучения полностью аналогичен стандартному BPE. Алгоритм итеративно находит наиболее частую пару соседних байтов или уже объединенных субсловных токенов и сливает их, добавляя новый токен в словарь. Например, если в тексте часто встречается слово «текст», которое в UTF-8 представляется последовательностью байтов, то алгоритм сначала объединит байты, соответствующие паре символов 'т' и 'е', затем, возможно, байты для 'к' и 'с', а потом и более крупные фрагменты.
3. **Обработка Unicode.** Многобайтовые символы UTF-8 просто рассматриваются как последовательности их составляющих байтов. Алгоритм BPE автоматически обучается объединять эти последовательности байтов обратно в осмысленные единицы, если они достаточно часто встречаются в корпусе.

Таким образом, byte-level BPE создает универсальную систему токенизации, способную обрабатывать любой текст на любом языке, включая код, специальные

символы и эмодзи, не требуя специальной предобработки или токена ‘<unk>’. Это делает его особенно мощным инструментом для современных многоязычных и многозадачных моделей.

1.3 Архитектура трансформер-кодировщик

Архитектура трансформер, предложенная в работе [6], изначально состояла из кодировщика (encoder) и декодировщика (decoder) для задач машинного перевода. Для задач векторизации, где требуется получить семантические представления для последовательности токенов, используется только кодирующая часть, называемая трансформер-кодировщиком.

Результатом работы трансформер-кодировщика $f_b(x_i, \alpha)$ является последовательность контекстуализированных векторных представлений $\mathbf{H}^{(L)} = (\mathbf{h}_1^{(L)}, \dots, \mathbf{h}_{n_i}^{(L)})$, по одному вектору для каждого входного токена. Модель можно представить как композицию двух основных компонентов: слоя векторизации токенов f_e и последовательности из L идентичных трансформер-блоков $f_{\text{TB}}^{(l)}$:

$$\mathbf{H}^{(L)} = f_b(x_i, \alpha) = f_{\text{TB}}^{(L)}\left(\dots f_{\text{TB}}^{(1)}(f_e(x_i, \alpha_e), \alpha_{\text{TB}}^{(1)}) \dots, \alpha_{\text{TB}}^{(L)}\right).$$

Рассмотрим каждый компонент подробнее.

Слой векторизации (Embedding Layer). На вход модели подается последовательность токенов $x_i = (w_{i,1}, \dots, w_{i,n_i})$. Сначала каждый токен отображается в вектор с помощью обучаемой матрицы эмбеддингов E . Процесс получения векторов начинается с токенизации, где входная текстовая последовательность преобразуется в последовательность целочисленных идентификаторов (id_1, \dots, id_{n_i}) в соответствии с предопределенным словарем \mathcal{V} размером $|\mathcal{V}|$. Каждый уникальный токен в словаре имеет свой уникальный числовой идентификатор.

Сама матрица эмбеддингов $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ по своей сути является таблицей поиска (lookup table), где каждой k -й строке, $k \in \{1, \dots, |\mathcal{V}|\}$, соответствует плотный вектор $\mathbf{v}_k \in \mathbb{R}^d$ для токена с идентификатором k . Таким образом, операция получения эмбеддинга для одного токена сводится к извлечению соответствующей строки из этой матрицы.

Формально эту операцию можно выразить через умножение матрицы E на one-hot-вектор. Для токена с идентификатором k создается вектор $\mathbf{e}_k \in \{0, 1\}^{|\mathcal{V}|}$, у которого k -я компонента равна 1, а все остальные — 0. Тогда векторное представление токена \mathbf{v}_k вычисляется как:

$$\mathbf{v}_k = \mathbf{e}_k^\top E.$$

Элементы матрицы E являются обучаемыми параметрами модели, которые настраиваются в процессе обучения методом обратного распространения ошибки. Применение этой операции ко всем токенам последовательности x_i порождает матрицу токеновых эмбеддингов $E(x_i) \in \mathbb{R}^{n_i \times d}$, которая и служит входом для дальнейших преобразований.

Поскольку модель не содержит рекуррентных или сверточных слоев, для внесения информации о порядке токенов используется матрица позиционных эмбеддингов P . Итоговое представление для входной последовательности, обозначаемое как $\mathbf{H}^{(0)}$, получается путем суммирования токеновых и позиционных эмбеддингов. В работе [6] предлагается также масштабировать токеновые эмбеддинги:

$$\mathbf{H}^{(0)} = E(x_i)\sqrt{d} + P,$$

где

$E \in \mathbb{R}^{|\mathcal{V}| \times d}$ — матрица эмбеддингов токенов, $|\mathcal{V}|$ — размер словаря.

d — размерность векторного представления.

$P \in \mathbb{R}^{n_{\max} \times d}$ — матрица позиционных эмбеддингов, n_{\max} — максимальная длина последовательности.

Компоненты матрицы позиционных эмбеддингов P вычисляются с использованием синусоидальных функций разной частоты:

$$P_{pos,2k} = \sin(pos/10000^{2k/d}), \quad P_{pos,2k+1} = \cos(pos/10000^{2k/d}),$$

где

pos — позиция токена в последовательности $(0, 1, \dots, n_i - 1)$.

k — индекс размерности вектора $(0, 1, \dots, d/2 - 1)$.

Такой выбор позволяет модели легко обучаться на относительных позициях, поскольку для любого фиксированного смещения j позиционный эмбеддинг P_{pos+j} может быть выражен как линейная функция от P_{pos} .

Трансформер-блок. Ядром архитектуры является трансформер-блок, который состоит из двух основных подслоев: механизма многоголового внимания (Multi-Head Attention) и полносвязной нейронной сети прямого распространения (Feed-Forward Network). Вокруг каждого из этих подслоев применяется остаточное соединение (residual connection) с последующей послойной нормализацией (Layer Normalization). Для l -го блока ($l = 1, \dots, L$) преобразование матрицы представлений $\mathbf{H}^{(l-1)}$ в $\mathbf{H}^{(l)}$ выглядит следующим образом:

$$\mathbf{Z}^{(l)} = \text{LayerNorm}(\mathbf{H}^{(l-1)} + \text{MHAtt}^{(l)}(\mathbf{H}^{(l-1)})),$$

$$\mathbf{H}^{(l)} = \text{LayerNorm}(\mathbf{Z}^{(l)} + \text{FF}^{(l)}(\mathbf{Z}^{(l)})).$$

Механизм внимания (Scaled Dot-Product Attention). Это ключевой компонент, который позволяет модели взвешивать важность различных токенов в последовательности при формировании представления для каждого конкретного токена. Входными данными для него служат три матрицы: запросов (Query, Q), ключей (Key, K) и значений (Value, V). В режиме самовнимания (self-attention), который используется в кодировщике, все три матрицы получаются путем линейного проецирования матрицы выходов предыдущего слоя $\mathbf{H}^{(l-1)}$ с помощью индивидуальных обучаемых весовых матриц для каждого типа проекции.

Формально, для l -го трансформер-блока эти преобразования записываются следующим образом:

$$\begin{aligned} Q &= \mathbf{H}^{(l-1)} W^{Q,(l)}, \\ K &= \mathbf{H}^{(l-1)} W^{K,(l)}, \\ V &= \mathbf{H}^{(l-1)} W^{V,(l)}, \end{aligned} \tag{1.1}$$

где

$\mathbf{H}^{(l-1)} \in \mathbb{R}^{n_i \times d}$ — матрица векторных представлений всех n_i токенов, полученная на выходе $(l-1)$ -го блока. Каждая строка этой матрицы — это вектор $\mathbf{h}_j^{(l-1)}$ для j -го токена.

$W^{Q,(l)}, W^{K,(l)} \in \mathbb{R}^{d \times d_k}$ — обучаемые матрицы весов для линейного проецирования в пространства запросов и ключей соответственно.

$W^{V,(l)} \in \mathbb{R}^{d \times d_v}$ — обучаемая матрица весов для линейного проецирования в пространство значений.

d, d_k, d_v — размерности пространств исходных представлений, ключей/запросов и значений. В архитектуре трансформер принято, что $d_k = d_v$.

Матрицы Q , K и V задают три представления одной и той же последовательности: в пространстве запросов, ключей и значений соответственно. На их основе вычисляется функция внимания:

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

Результатом операции $\text{Attn}(Q, K, V)$ является матрица из пространства $\mathbb{R}^{n_i \times d_v}$, содержащая обновленные представления всех токенов. Представление каждого токена строится как взвешенная сумма векторов из матрицы значений V . Веса определяются сходством вектора запроса данного токена с векторами ключей остальных токенов последовательности, которое вычисляется с помощью скалярного произведения. Множитель $1/\sqrt{d_k}$ используется для масштабирования значений перед применением функции softmax и повышает устойчивость обучения при больших размерностях d_k .

Многоголовое внимание (Multi-Head Attention). Вместо одного механизма внимания в трансформере параллельно используются H «голов» внимания. Для каждой головы входные представления $\mathbf{H}^{(l-1)}$ проецируются с помощью собственного набора весовых матриц, после чего вычисляется функция внимания. Полученные представления объединяются конкатенацией и проходят заключительное линейное преобразование:

$$\begin{aligned} \text{head}_h &= \text{Attn}(\mathbf{H}^{(l-1)}W_h^{Q,(l)}, \mathbf{H}^{(l-1)}W_h^{K,(l)}, \mathbf{H}^{(l-1)}W_h^{V,(l)}), \\ \text{MHAtt}^{(l)}(\mathbf{H}^{(l-1)}) &= \text{Concat}(\text{head}_1, \dots, \text{head}_H)W^{O,(l)}, \end{aligned}$$

где $h = 1, \dots, H$ — номер головы. Такая организация позволяет модели одновременно учитывать разные типы связей между токенами и извлекать информацию из нескольких подпространств представлений.

Полносвязная сеть (Feed-Forward Network). Этот подслой применяется независимо к представлению каждого токена и состоит из двух линейных преобразований, между которыми находится нелинейная функция активации. В качестве функции активации используется ReLU (Rectified Linear Unit) [48], определяемая как $\text{ReLU}(z) = \max(0, z)$ и применяемая поэлементно. Преобразование для матрицы \mathbf{Z} имеет вид:

$$\text{FF}^{(l)}(\mathbf{Z}) = \text{ReLU}(\mathbf{Z}W_1^{(l)} + \mathbf{b}_1^{(l)})W_2^{(l)} + \mathbf{b}_2^{(l)}.$$

Послойная нормализация (Layer Normalization) [49]. Послойная нормализация применяется для стабилизации обучения. Для каждого токена она

выполняется отдельно по компонентам его векторного представления:

$$\text{LayerNorm}(\mathbf{x}) = \gamma \frac{\mathbf{x} - \mu(\mathbf{x})}{\sqrt{\sigma^2(\mathbf{x}) + \varepsilon}} + \beta,$$

где $\mu(\mathbf{x})$ и $\sigma^2(\mathbf{x})$ — среднее и дисперсия по компонентам вектора \mathbf{x} , а γ и β — обучаемые параметры масштаба и сдвига.

1.4 Предобучение моделей на основе архитектуры трансформер

Модель **BERT** (Bidirectional Encoder Representations from Transformers) [18] стала революционным шагом в обработке естественного языка. Ее архитектура представляет собой описанный в разделе 1.3 трансформер-кодировщик. Ключевое нововведение BERT заключается в методологии его **предобучения** на огромных объемах размеченных текстовых данных, включающей две задачи: маскированное языковое моделирование (MLM) и предсказание следующего предложения (NSP). В качестве обучающих данных использовались большие объемы англоязычных текстов, а именно, корпус BooksCorpus [18] (около 800 млн слов) и англоязычная Википедия (2.5 млрд слов). Совокупный объем данных (около 16 Гб текста) и их тематическое разнообразие (художественная литература, научные и энциклопедические статьи) позволили модели изучить широкий спектр языковых явлений и закономерностей, создав универсальные представления.

В отличие от предыдущих моделей, таких как GPT, которые были однонаправленными, BERT является глубоко двунаправленной моделью, то есть при построении представления каждого токена он одновременно учитывает как левый, так и правый контекст на всех слоях.

Маскированное языковое моделирование (Masked Language Modeling, MLM). Эта задача позволяет обучать двунаправленные представления. Во входной последовательности токенов x_i случайным образом выбирается подмножество позиций M_i (15%) для маскирования. Для каждой позиции j из этого

подмножества токенов $w_{i,j}$ заменяется по следующему правилу:

$$w'_{i,j} = \begin{cases} [\text{MASK}], & \text{с вероятностью } 80\%; \\ u \sim \mathcal{U}(\mathcal{V}), & \text{с вероятностью } 10\%; \\ w_{i,j}, & \text{с вероятностью } 10\%. \end{cases}$$

Формально, схема маскирования описывается как

$$m_{i,j} \sim \text{Bernoulli}(0.15), \quad r \sim \mathcal{U}(0,1),$$

$$w'_{i,j} = \begin{cases} [\text{MASK}], & m_{i,j} = 1 \text{ и } r < 0.8; \\ u \sim \mathcal{U}(\mathcal{V}), & m_{i,j} = 1 \text{ и } 0.8 \leq r < 0.9; \\ w_{i,j}, & \text{иначе,} \end{cases}$$

где $m_{i,j}$ — индикатор маскирования, r — равномерно распределенная случайная величина, $\mathcal{U}(\mathcal{V})$ — равномерное распределение по словарю \mathcal{V} . В оригинальной реализации BERT [18] используется **статическое маскирование**: для каждого обучающего примера маска создается однократно и остается неизменной на протяжении всех эпох обучения. Задача модели — на основе маскированной последовательности $x'_i = (w'_{i,1}, \dots, w'_{i,n_i})$ предсказать исходные токены в позициях $j \in M_i = \{j \mid m_{i,j} = 1\}$. Для этого выходы кодировщика $\mathbf{h}_j^{(L)}$ подаются на классификационный слой:

$$p_{ij} = \text{softmax}(\mathbf{h}_j^{(L)} \mathbf{W}_{\text{MLM}}^\top + \mathbf{b}_{\text{MLM}}),$$

где $\mathbf{W}_{\text{MLM}} \in \mathbb{R}^{|\mathcal{V}| \times d}$ и $\mathbf{b}_{\text{MLM}} \in \mathbb{R}^{|\mathcal{V}|}$ — обучаемые параметры. Функция потерь MLM минимизирует перекрестную энтропию по маскированным токенам:

$$\mathcal{L}_{\text{MLM}} = -\frac{1}{\sum_{i=1}^B |M_i|} \sum_{i=1}^B \sum_{j \in M_i} \log p_{ij}[y_{ij}] \longrightarrow \min,$$

где B — размер мини-выборки, y_{ij} — индекс истинного токена в позиции j последовательности x_i .

Предсказание следующего предложения (Next Sentence Prediction, NSP). Эта задача была введена для того, чтобы модель могла улавливать связи между предложениями, что важно для таких задач, как ответы на вопросы и определение логического следования. Для формирования обучающей выборки создаются пары

предложений (A, B) . В 50% случаев предложение B является фактическим следующим предложением за A в исходном корпусе (метка ‘IsNext’), а в остальных 50% случаев B является случайным предложением из корпуса (метка ‘NotNext’).

На вход модели подается конкатенированная последовательность: ‘[CLS]’ A ‘[SEP]’ B ‘[SEP]’. Для решения этой задачи бинарной классификации используется векторное представление специального токена [CLS], а именно вектор $\mathbf{h}_{\text{CLS}}^{(L)}$, который подается на простой классификатор:

$$p_{\text{NSP}} = \text{softmax}(\mathbf{h}_{\text{CLS}}^{(L)} \mathbf{W}_{\text{NSP}}^{\top} + \mathbf{b}_{\text{NSP}}).$$

Функция потерь \mathcal{L}_{NSP} вычисляется как стандартная перекрестная энтропия для бинарной классификации.

Совместное обучение. Процесс предобучения BERT заключается в одновременной минимизации суммарной функции потерь по обоим задачам. Обучение ведется на мини-выборках (mini-batches) — небольших случайных подмножествах данных. Итоговая функция потерь, градиент которой вычисляется на каждом шаге, представляет собой сумму функций потерь MLM и NSP:

$$\mathcal{L}_{\text{total}}(\alpha, \theta_{\text{MLM}}, \theta_{\text{NSP}}) = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{NSP}} \longrightarrow \min_{\alpha, \theta_{\text{MLM}}, \theta_{\text{NSP}}},$$

где α обозначает параметры трансформер-кодировщика, а $\theta_{\text{MLM}} = \{\mathbf{W}_{\text{MLM}}, \mathbf{b}_{\text{MLM}}\}$ и $\theta_{\text{NSP}} = \{\mathbf{W}_{\text{NSP}}, \mathbf{b}_{\text{NSP}}\}$ — параметры классификационных слоев для каждой из задач.

Модель RoBERTa. Развивая идеи BERT, модель RoBERTa (A Robustly Optimized BERT Approach) [46] не вводит новой архитектуры, а представляет собой тщательное исследование и оптимизацию процесса обучения BERT. Авторы показали, что производительность BERT сильно зависит от гиперпараметров и стратегии обучения. Ключевые отличия RoBERTa от BERT:

- **Динамическое маскирование.** В RoBERTa маска генерируется заново для каждой последовательности при каждой ее подаче в модель, что существенно увеличивает разнообразие обучающих данных.
- **Отказ от задачи NSP.** Авторы показали, что исключение этой задачи улучшает качество итоговых представлений на многих задачах.
- **Обучение на больших батчах и данных.** RoBERTa обучалась на значительно больших мини-выборках (размером до 8000 примеров) и на большем объеме текстовых данных (160 ГБ текста против 16 ГБ у BERT).

- **Токенизация.** Вместо токенизатора WordPiece [50], RoBERTa использует алгоритм byte-level Byte-Pair Encoding (BPE) [43] с большим размером словаря (50 тыс. токенов против 30 тыс. токенов у модели BERT), что позволяет избежать токенов неизвестных слов.

Эти изменения позволили моделям RoBERTa существенно превзойти по качеству оригинальные модели BERT на широком спектре задач и установили новый стандарт для предобученных трансформер-кодировщиков.

1.5 Контрастивное дообучение для векторных представлений текста

Предобученные трансформер-кодировщики, такие как BERT [18], продемонстрировали высокую эффективность в задачах классификации для отдельных текстов или пар текстов, задачах ответа на вопросы и тегирования слов. Однако практические задачи, такие как семантический поиск, кластеризация или обнаружение дубликатов, требуют решения другой, хотя и связанной, проблемы: для данного текста-запроса x_q найти наиболее семантически близкий текст в большой коллекции $\mathcal{D} = \{x_i\}_{i=1}^N$.

Прямолинейный подход, следующий из оригинальной архитектуры BERT, заключается в использовании так называемой архитектуры перекрестного кодировщика (cross-encoder). В этой схеме пара текстов, запрос x_q и документ x_i , конкатенируется в одну общую последовательность $x_{q,i}$, разделенную специальным токеном [SEP]:

$$x_{q,i} = ([\text{CLS}], x_q, [\text{SEP}], x_i, [\text{SEP}]).$$

Эта последовательность подается на вход трансформер-кодировщика $f_b(x, \alpha)$. Выходной вектор $\mathbf{h}_{\text{CLS}}^{(L)}$, соответствующий специальному токеному [CLS], агрегирует информацию обо всей паре и используется для предсказания оценки их близости $s_{q,i}$ с помощью простого регрессионного слоя с параметрами $\theta_{reg} = \{\mathbf{w}, b\}$:

$$\hat{s}_{q,i} = \mathbf{w}^\top \mathbf{h}_{\text{CLS}}^{(L)} + b,$$

где

$\mathbf{h}_{\text{CLS}}^{(L)} \in \mathbb{R}^d$ — выходной вектор кодировщика для токена [CLS] при подаче на вход последовательности $x_{q,i}$.

$w \in \mathbb{R}^d, b \in \mathbb{R}$ — обучаемые параметры регрессионного слоя.

Процесс дообучения такой модели заключается в совместном подборе параметров кодировщика α и регрессионного слоя θ_{reg} путем минимизации функции потерь на специализированном наборе данных, на корпусе семантической близости текстов (Semantic Textual Similarity, STS) [51]. Для каждой обучающей пары (x_q, x_i) с известной экспертной оценкой близости $s_{q,i}$ минимизируется средне-квадратичная ошибка:

$$\mathcal{L}(\alpha, \theta_{reg}) = \sum_{(x_q, x_i, s) \in \text{STS}} (\hat{s}_{q,i} - s_{q,i})^2 \longrightarrow \min_{\alpha, \theta_{reg}} .$$

Несмотря на то, что такой подход позволяет достичь высокой точности, его вычислительная сложность делает его неприменимым для большинства реальных сценариев. Поиск наиболее похожего текста для одного запроса x_q в коллекции из N документов потребует выполнения N прямых проходов через модель трансформера.

Фундаментальное ограничение перекрестного кодировщика состоит в том, что он не порождает независимого векторного представления для каждого отдельного текста. Оценка близости вычисляется «с нуля» для каждой новой пары. Подход, лишённый этого вычислительного недостатка, заключается в построении архитектуры, которая позволяет заранее отобразить каждый текст x_i из коллекции \mathcal{D} в его собственный вектор $v_i = f(x_i, \alpha)$. В этом случае вся коллекция текстов может быть векторизована однократно. Процесс поиска для нового запроса x_q сведется к вычислению его вектора v_q и последующему поиску ближайшего соседа в уже готовом векторном пространстве с помощью мер близости, например косинусной. Этот подход, известный как архитектура с двумя кодировщиками (bi-encoder) или сиамская сеть (siamese network), кардинально снижает сложность поиска. Он требует лишь одного применения модели для векторизации запроса, в отличие от N применений у перекрестного кодировщика.

Следовательно, для создания практичных и масштабируемых систем семантического поиска необходимо отказаться от схемы перекрестного кодировщика в пользу архитектуры, способной генерировать высококачественные независимые векторные представления текстов. Далее рассмотрим, как можно дообучить модели типа BERT для решения именно этой задачи.

Несмотря на то, что предобученные на задаче маскированного языкового моделирования (MLM) трансформер-кодировщики, такие как BERT [18] и

RoBERTa [46], способны генерировать контекстуализированные векторные представления для каждого токена, они изначально не оптимизированы для задачи получения единого, семантически осмысленного вектора для целого предложения или документа. Как было показано в разделе 1.3, результатом работы кодировщика является матрица векторов $\mathbf{H}^{(L)} \in \mathbb{R}^{n_i \times d}$. Для решения прикладных задач, таких как семантический поиск, кластеризация или верификация фактов, необходимо агрегировать эту последовательность векторов в единственный вектор $\mathbf{v}_i \in \mathbb{R}^d$. Этот шаг, однако, не является тривиальным и вскрывает фундаментальное несоответствие между целями предобучения и целями векторизации.

Простейшие стратегии агрегации, или пулинга, включают усреднение векторов всех токенов (Mean Pooling) или использование выходного вектора специального токена [CLS] (CLS Pooling), который добавляется в начало каждой последовательности. В первом случае вектор документа вычисляется как:

$$\mathbf{v}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{h}_j^{(L)},$$

где $\mathbf{h}_j^{(L)}$ — векторное представление j -го токена на выходе последнего слоя кодировщика. Во втором случае в качестве вектора документа используется вектор, соответствующий позиции токена [CLS]:

$$\mathbf{v}_i = \mathbf{h}_{\text{CLS}}^{(L)}.$$

Ключевая проблема этих подходов заключается в том, что ни один из этапов предобучения BERT-подобных моделей не ставит своей явной целью формирование семантически структурированного пространства векторов на уровне предложений. Функция потерь MLM, как было описано в разделе 1.4, оптимизирует параметры модели для предсказания маскированных токенов, но не содержит компоненты, которая бы сближала векторы семантически похожих предложений и отдаляла векторы разных по смыслу. Задача предсказания следующего предложения (NSP) в оригинальном BERT формально обучала вектор $\mathbf{h}_{\text{CLS}}^{(L)}$ для бинарной классификации, однако, как показали авторы RoBERTa [46], эта задача является слишком грубой и её исключение из процесса предобучения зачастую приводит к улучшению качества итоговых представлений. Таким образом, в современных моделях, отказавшихся от NSP, вектор [CLS] не несет специальной семантической нагрузки на уровне предложения.

Эмпирические исследования подтверждают теоретические недостатки наивных стратегий агрегации. В работе по модели Sentence-BERT [19] было экспериментально показано, что векторные представления, полученные напрямую из BERT без специального дообучения, непригодны для задач семантического поиска. Качество таких векторов оказывается крайне низким, уступая даже более простым неконтекстуализированным моделям, таким как усредненные векторы GloVe[16].

Для преодоления этих ограничений необходим дополнительный этап обучения — дообучение (fine-tuning), направленный на формирование такой структуры векторного пространства, в которой геометрическая близость векторов напрямую соответствует семантической близости текстов. Идея состоит в том, чтобы оптимизировать параметры модели α с помощью новой функции потерь, которая явно поощряет нужное расположение векторов. Для этого используются обучающие наборы данных, состоящие из пар или троек текстов с известными семантическими отношениями. Архитектурно эта задача решается с помощью сиамских (siamese) сетей, в которых два или более экземпляра одной и той же модели-кодировщика $f(x, \alpha)$ с общими параметрами α параллельно обрабатывают входные тексты. Использование общих весов гарантирует, что все тексты отображаются в единое векторное пространство, где их можно сравнивать.

Существует множество подходов к реализации контрастивного дообучения, которые можно разделить на две основные группы в зависимости от структуры используемых обучающих примеров: методы, основанные на тройках текстов, и методы, работающие с парами и формирующие отрицательные примеры динамически.

Первая группа методов оперирует тройками (x_a, x_p, x_n) , где x_a — опорный текст («якорь», anchor), x_p — семантически схожий с ним текст («положительный пример», positive), а x_n — семантически непохожий текст («отрицательный пример», negative). Цель обучения — минимизировать расстояние между векторами якоря и положительного примера, одновременно максимизируя расстояние между векторами якоря и отрицательного примера. Классическим представителем этого семейства является триплетная функция потерь (Triplet Loss), представленная в работе [52] и успешно адаптированная для текстов в работе [19]. Она формализует требование, чтобы расстояние до отрицательного примера было больше расстояния до положительного как минимум на некоторую величину $\epsilon > 0$.

$$\mathcal{L}_{\text{triplet}} = \sum_{i=1}^B \max(0, d(\mathbf{v}_{a,i}, \mathbf{v}_{p,i}) - d(\mathbf{v}_{a,i}, \mathbf{v}_{n,i}) + \varepsilon) \longrightarrow \min_{\alpha}, \quad (1.2)$$

где

B — размер мини-выборки (mini-batch).

$\mathbf{v}_{a,i}, \mathbf{v}_{p,i}, \mathbf{v}_{n,i}$ — векторные представления якоря, положительного и отрицательного примеров из i -й тройки, полученные с помощью кодировщика $f(x, \alpha)$.

$d(\cdot, \cdot)$ — функция расстояния, например, евклидово расстояние.

$\varepsilon > 0$ — гиперпараметр, задающий минимально допустимый зазор (margin).

Основная сложность этого подхода заключается в необходимости формирования качественных троек, в частности, в подборе «трудных» отрицательных примеров (hard negatives), которые не слишком далеки от якоря и заставляют модель изучать тонкие семантические различия.

Вторая, более современная группа методов использует для обучения только положительные пары (x_a, x_p) и формирует отрицательные примеры «на лету» из других примеров в той же мини-выборке. Такой подход, известный как обучение с отрицательными примерами из выборки (in-batch negatives), значительно эффективнее, так как не требует явного конструирования троек. Одним из наиболее распространенных вариантов является функция потерь InfoNCE (Noise-Contrastive Estimation) [20]. Идея состоит в том, чтобы научиться отличать «настоящий» положительный пример от «шумовых» отрицательных примеров. В контексте векторизации текстов это сводится к различению истинной положительной пары от пар, составленных с отрицательными примерами. Для мини-выборки из B положительных пар $\{(x_{a,i}, x_{p,i})\}_{i=1}^B$ функция потерь имеет вид:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{e^{s(\mathbf{v}_{a,i}, \mathbf{v}_{p,i})/\tau}}{e^{s(\mathbf{v}_{a,i}, \mathbf{v}_{p,i})/\tau} + \sum_{j=1, j \neq i}^B e^{s(\mathbf{v}_{a,i}, \mathbf{v}_{p,j})/\tau}} \longrightarrow \min_{\alpha}, \quad (1.3)$$

где

$\mathbf{v}_{a,i}, \mathbf{v}_{p,i}$ — векторы для i -й положительной пары.

$\mathbf{v}_{p,j}$ — вектор для j -го примера из мини-выборки, используемого в качестве отрицательного для i -й пары.

$s(\cdot, \cdot)$ — функция оценки близости, как правило, косинусная близость.

$\tau > 0$ — температурный коэффициент, который масштабирует распределение оценок. Малые значения τ делают распределение более резким, заставляя модель сильнее различать примеры.

Ключевым теоретическим свойством функции InfoNCE является то, что ее минимизация эквивалентна максимизации нижней оценки взаимной информации между представлениями якоря и положительного примера [20]. Это придает подходу прочное теоретическое основание.

1.6 Применение и оценка моделей в научной области

После того как модель векторного представления $f(x, \alpha)$ прошла этапы предобучения и дообучения, ее основной целью становится применение для решения прикладных задач. Ценность современных моделей, основанных на архитектуре трансформер, заключается в том, что масштабное предобучение на гигантских текстовых корпусах позволяет им накопить фундаментальные знания о языке, синтаксисе и семантике. Это знание, закодированное в параметрах α , может быть эффективно перенесено на новые, в том числе узкоспециализированные, задачи, даже при наличии очень малого количества размеченных данных или вовсе без них. Парадигмы применения в режиме «нулевого выстрела» или с обучением легковесной «головы», описанные в следующем разделе, являются ключевыми инструментами для такого переноса знаний и оценки его успешности.

Существует множество мощных моделей общего назначения, обученных на гетерогенных корпусах огромного объема, таких как веб-страницы (например, Common Crawl) и Wikipedia. Такие модели демонстрируют способность к обобщению и служат надежной отправной точкой для широкого круга задач. Однако их эффективность может снижаться при переносе в предметные области, лексические, стилистические и структурные особенности которых существенно отличаются от усредненного «общего» языка обучающих данных.

Одним из наиболее ярких примеров такой узкоспециализированной области является научный дискурс [53]. Научные тексты обладают рядом уникальных характеристик, отличающих их от новостных статей, художественной литературы или веб-контента.

- **Лексика:** Научные тексты насыщены узкоспециализированной терминологией, обилием акронимов и устойчивых словосочетаний (коллокаций), которые редко встречаются в общеупотребительной лексике.
- **Структура:** Научная публикация имеет устойчивую структуру, обычно включающую аннотацию, введение, описание методов, результаты и заключение. Положение фрагмента в этой структуре связано с его функцией в тексте и влияет на интерпретацию содержащейся в нем информации.
- **Синтаксис и стиль:** Для научных текстов характерны развернутые синтаксические конструкции, частое использование страдательного залога и безличных форм. Изложение ориентировано на точность, объективность и однозначность формулировок, поэтому в нем ограничено использование образных и многозначных выражений.
- **Семантические связи:** Научные тексты отличаются высокой информационной плотностью и сложными логическими отношениями между утверждениями. Особое значение имеют цитирования, которые связывают публикации между собой и задают научный контекст рассматриваемой работы [22].

Перечисленные лексические, структурные и семантические особенности приводят к доменному сдвигу (domain shift) [54]. В результате модели общего назначения могут некорректно учитывать контекст, значимость терминов и связи между смысловыми частями научной статьи.

Таким образом, универсальные модели могут использоваться как основа для решения задач обработки научных текстов, однако для повышения качества обычно требуется доменная адаптация. По этой причине были разработаны специализированные модели, которые либо предобучаются на больших корпусах научных статей, либо дополнительно дообучаются на таких данных. Такой подход позволяет получать векторные представления, лучше отражающие особенности научного текста. Далее рассматриваются модели, созданные для научного домена, и способы их оценки.

Модель SPECTER

Одной из первых моделей, реализующих доменный подход к векторизации научных текстов, стала **SPECTER** (Scientific Paper Embeddings using Citation-informed Transformers) [22]. Ключевая идея авторов заключается в использовании естественного и семантически богатого сигнала, присущего научной среде - цитирований. В отличие от общих задач семантической близости, где пары похожих текстов часто подбираются эвристически или с помощью разметки методами краудсорсинга, граф цитирований предоставляет масштабный и объективный источник информации о смысловой связи между документами.

Модель SPECTER использует в качестве основы трансформер-кодировщик SciBERT [21], предобученный на научных текстах, и дообучает его с помощью триплетной функции потерь. В качестве входных данных для каждой научной статьи x_i используется конкатенация ее заголовка и аннотации. Обучающие тройки (x_q, x_p, x_n) формируются непосредственно из графа цитирований:

- **Опорный текст (якорь) x_q** : произвольная научная статья.
- **Положительный пример x_p** : статья, которую цитирует якорь x_q .
- **Отрицательный пример x_n** : статья, которую якорь x_q не цитирует.

Дообучение модели выполняется путем минимизации триплетной функции потерь, аналогичной выражению (1.2). Расстояние $d(\cdot, \cdot)$ в данном случае задается евклидовым расстоянием, то есть L_2 -нормой между векторными представлениями статей.

$$\mathcal{L}_{\text{SPECTER}} = \max(0, \|\mathbf{v}_q - \mathbf{v}_p\|_2 - \|\mathbf{v}_q - \mathbf{v}_n\|_2 + m) \longrightarrow \min_{\alpha},$$

где $\mathbf{v}_q, \mathbf{v}_p, \mathbf{v}_n$ — векторные представления, полученные из модели $f(x, \alpha)$, а m — гиперпараметр зазора.

Отдельное место в SPECTER занимает выбор отрицательных примеров. Помимо случайно выбранных документов из корпуса, используются «трудные отрицательные примеры» (hard negatives). Для якоря x_q такими примерами считаются статьи, которые цитируются положительным примером x_p , но не цитируются самим якорем. Иными словами, если в графе цитирований есть цепочка $x_q \rightarrow x_p \rightarrow x_{n'}$, но нет прямого ребра $x_q \not\rightarrow x_{n'}$, то статья $x_{n'}$ используется как трудный отрицательный пример. Такие документы близки к якорю по тематике,

поэтому модель вынуждена различать более тонкие смысловые различия между публикациями.

SPECTER2 и мультимедийное обучение

Дальнейшим развитием идей доменной адаптации стала модель **SPECTER2** [23]. Ее разработка основана на наблюдении, что одно универсальное представление документа, даже обученное с использованием сигнала цитирования, не всегда одинаково хорошо подходит для разных прикладных задач. Классификация, поиск и регрессия предъявляют разные требования к структуре векторного пространства. В задачах классификации важно, чтобы документы одного класса образовывали хорошо разделяемые группы, а в задачах поиска и ранжирования особенно значимо сохранение локальной близости и относительных расстояний между документами.

SPECTER2 отказывается от принципа «один документ — один вектор» и использует **мультимедийное обучение**. Для одного и того же документа модель формирует несколько представлений, каждое из которых ориентировано на отдельный формат задач: классификацию (Classification, CLF), регрессию (Regression, RGN), поиск по близости (Proximity, PRX) и поиск по произвольному запросу (Ad-hoc Search, SRCH).

Для реализации этого подхода авторы исследовали два основных механизма.

- **Управляющие коды (Control Codes)**. Этот метод заключается в добавлении в начало входной последовательности специального токена, соответствующего целевому формату задачи. Например, для получения вектора, оптимизированного для классификации, на вход модели подается последовательность $x'_i = ([\text{CLF}], x_i)$, где x_i — конкатенированные заголовок и аннотация. В качестве итогового представления документа для данного формата используется выходной вектор трансформера, соответствующий позиции этого управляющего токена. Этот подход является вычислительно эффективным, так как требует лишь добавления нескольких новых токенов в словарь без изменения архитектуры самой модели.

- **Адаптеры (Adapters)**. Этот механизм относится к техникам параметро-эффективного дообучения (Parameter-Efficient Fine-Tuning, PEFT) [55]. В каждый слой замороженной базовой модели-кодировщика встраиваются небольшие, легковесные нейросетевые модули — адаптеры. Для каждого формата задач создается свой, независимый набор адаптеров. В процессе обучения настраиваются только параметры этих адаптеров, в то время как основная часть модели остается неизменной. Этот подход позволяет более гибко специализировать модель под каждый формат, хотя и является более сложным в реализации.

Процесс обучения SPECTER2 представляет собой многозадачное обучение с гетерогенной функцией потерь. Обучение происходит на мини-выборках, каждая из которых относится к одному из четырех форматов. В зависимости от формата выборки для вычисления градиента используется своя функция потерь:

- для задач **классификации** — перекрестная энтропия (Cross-Entropy Loss):

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B} \sum_{i=1}^B \sum_{k=1}^K y_{ik} \log(p_{ik}) \longrightarrow \min,$$

где B — размер мини-выборки, K — число классов, y_{ik} — индикатор истинного класса (1, если i -й пример принадлежит классу k , и 0 иначе), а p_{ik} — предсказанная моделью вероятность принадлежности i -го примера к классу k .

- для задач **регрессии** — среднеквадратичная ошибка (Mean Squared Error, MSE):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{B} \sum_{i=1}^B (\hat{y}_i - y_i)^2 \longrightarrow \min,$$

где \hat{y}_i — предсказанное моделью значение для i -го примера, а y_i — истинное значение.

- для задач **поиска** (proximity и ad-hoc) — триплетная функция потерь, определенная в выражении (1.2).

SciNCL: Контрастивное обучение на соседях в графе цитирований

Несмотря на успех модели SPECTER, ее подход к формированию обучающих троек имеет фундаментальное ограничение: он основан на **дискретном сигнале** — факте наличия или отсутствия прямой цитаты. Такой бинарный подход является упрощением, поскольку семантическая близость — это непрерывная величина. Две статьи могут быть очень близки по теме, но не цитировать друг друга, в то время как некоторые цитаты могут быть сделаны из вежливости или для указания на контраргументы, не подразумевая сильной семантической связи [56]. Кроме того, однонаправленность сигнала (рассматриваются только исходящие цитаты) может приводить к «коллизиям», когда одна и та же пара статей в разных тройках может быть интерпретирована и как положительная, и как отрицательная, что вносит шум в процесс обучения.

Для решения этих проблем была предложена модель SciNCL (Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings) [24]. Ключевое нововведение SciNCL — переход от дискретного сигнала цитирования к мере близости, полученной из структуры всего графа цитирований. Это позволяет реализовать более гибкую и теоретически обоснованную стратегию сэмплирования обучающих примеров.

Подход SciNCL состоит из двух этапов. Сначала на всем графе цитирований научных статей с помощью алгоритма PyTorch BigGraph [57] обучается отдельная графовая модель векторного представления. В результате строится вспомогательное пространство векторов $\{c_i\}_{i=1}^N$, где каждый вектор c_i кодирует структурное положение статьи x_i в графе цитирований, а расстояние между векторами служит мерой близости между статьями. На втором этапе основная модель-кодировщик $f(x, \alpha)$ дообучается с помощью триплетной функции потерь. В отличие от SPECTER, положительные и отрицательные примеры для опорной статьи x_q выбираются с учетом ее соседей во вспомогательном пространстве векторов $\{c_i\}$.

Такая двухэтапная схема дает возможность управлять сложностью обучающих примеров. Положительные примеры x_p выбираются из ближайшего окружения якоря x_q во вспомогательном пространстве. Благодаря этому в обучающую выборку попадают не только очевидно близкие соседи, но и «трудные положительные примеры» (hard positives), то есть статьи, достаточно близкие к

якорю, чтобы считаться релевантными, но не совпадающие с ним по содержанию. Отрицательные примеры x_n также могут различаться по сложности: от случайных статей, удаленных от якоря, до «трудных» отрицательных примеров, расположенных близко к границе области положительных примеров.

Для разделения областей выбора положительных и отрицательных примеров вводится «отступ, определяемый примером» (sample-induced margin). Этот гиперпараметр задает во вспомогательном пространстве промежуточную область, из которой примеры не выбираются. Такой отступ снижает вероятность коллизий, когда близкие по структуре статьи получают противоречивые роли в разных обучающих тройках, и делает обучающий сигнал более согласованным.

Итоговая функция потерь идентична той, что используется в SPECTER, — это триплетная функция потерь (1.2). Однако ее эффективность кардинально повышается за счет более совершенного метода формирования троек. Этот метод позволяет модели обучаться на более гладком и непрерывном семантическом сигнале, что приводит к формированию более качественных и робастных векторных представлений научных текстов.

1.7 Оценка качества моделей векторного представления текстов

Создание эффективных моделей векторного представления текстов требует не только разработки новых архитектур и методов обучения, но и наличия надежных инструментов для их объективной оценки и сопоставления. Стандартизированные наборы данных и согласованные меры качества, объединенные в так называемые бенчмарки, играют в этом процессе ключевую роль. Они позволяют воспроизводимо и непредвзято сравнивать различные подходы, выявлять их сильные и слабые стороны и направлять дальнейшие исследования. В данной главе рассматриваются основные бенчмарки и меры качества, используемые для оценки моделей в области обработки научных текстов, начиная с основополагающих работ и заканчивая современными комплексными наборами задач.

1.7.1 Бенчмарк SciDocs

Одним из первых комплексных инструментов, разработанных специально для оценки качества векторных представлений научных документов, стал бенчмарк **SciDocs** [22]. Его создание было мотивировано тем, что существующие на тот момент наборы данных для оценки семантической близости были либо слишком малы, либо задачи на них были практически решены (меры качества достигали 99%), что не позволяло адекватно сравнивать новые, более мощные модели. SciDocs был предложен как более сложный и разносторонний набор задач, призванный оценить способность моделей к обобщению.

Ключевой принцип, заложенный в SciDocs, — это оценка моделей в режиме «как есть» (as features), то есть без дополнительного дообучения параметров самого кодировщика на задачах бенчмарка. Полученные векторные представления подаются на вход простым, легковесным моделям (например, линейному классификатору), и оценивается именно качество этих представлений как признаков для решения целевых задач. Бенчмарк включает семь задач, сгруппированных в четыре категории.

Классификация документов. Эта категория содержит две задачи, проверяющие, насколько хорошо векторное представление кодирует тематическую принадлежность документа. Качество решения в обеих задачах оценивается с помощью **макро-усредненной F1-меры** (Macro F1-score), которая является стандартной мерой для многоклассовой классификации, устойчивой к дисбалансу классов.

- **MeSH Classification.** Задача классификации медицинских статей по 11 высокоуровневым классам заболеваний (например, «сердечно-сосудистые заболевания», «диабет») из тезауруса Medical Subject Headings (MeSH) [58].
- **Paper Topic Classification.** Задача классификации статей по 19 предметным областям первого уровня (например, «Физика», «Математика», «Информатика») из иерархии Microsoft Academic Graph (MAG) [59].

Предсказание цитирований. Эти задачи напрямую оценивают способность модели воспроизводить семантический сигнал, заложенный в цитированиях. Обе задачи сформулированы как задачи ранжирования, и их качество оценивается

с помощью стандартных для информационного поиска мер: **средней точности (Mean Average Precision, MAP)** и **нормализованного дисконтированного совокупного выигрыша (Normalized Discounted Cumulative Gain, nDCG)**.

- **Direct Citations.** Для данной статьи-запроса необходимо ранжировать предложенный набор статей-кандидатов таким образом, чтобы цитируемые ею статьи оказались выше в списке, чем случайно выбранные нецитируемые.
- **Co-Citations.** Задача аналогична предыдущей, но вместо прямых цитирований требуется предсказать статьи, которые часто цитируются совместно с запросной статьей, что является сильным индикатором тематической близости.

Анализ пользовательской активности. В этой категории в качестве косвенного индикатора семантической близости используется поведение пользователей на крупном научном поисковом портале. Поскольку эти задачи также являются задачами ранжирования, для их оценки применяются те же меры качества: **MAP** и **nDCG**.

- **Co-Views.** Модель должна ранжировать статьи, которые пользователи часто просматривали в рамках одной и той же поисковой сессии, выше, чем случайные статьи.
- **Co-Reads.** Более сильный сигнал, основанный на кликах по PDF-файлам. Предполагается, что если пользователи скачивают несколько статей в одной сессии, эти статьи с высокой вероятностью тесно связаны по теме.

Рекомендация статей. Наиболее прикладная задача, в которой векторное представление используется не изолированно, а как один из признаков (feature) в существующей промышленной рекомендательной системе. В качестве мер качества используются **нормализованный дисконтированный совокупный выигрыш (nDCG)** и **точность на первой позиции (Precision@1)**, скорректированные с учетом смещения. Для корректной оценки используется механизм корректировки на основе оценок склонности (propensity scores), чтобы нивелировать смещение, возникающее из-за того, что пользователи чаще кликают на элементы вверху списка.

Таким образом, SciDocs предоставляет разносторонний набор тестов, проверяющих различные аспекты качества векторных представлений, от те-

матической классификации до воспроизведения сложных семантических связей, отраженных в поведении научного сообщества.

1.7.2 Бенчмарк SciRepEval

В бенчмарке SciDocs был выявлен ряд ограничений, послуживших предпосылкой для создания более совершенных инструментов оценки. Было показано, что задачи в SciDocs недостаточно разнообразны, а некоторые из них, в частности, задачи поиска со случайно выбранными отрицательными примерами, оказались слишком простыми для современных моделей. Важным недостатком также являлось то, что все задачи в SciDocs были предназначены исключительно для оценки, что не позволяло исследовать эффекты многозадачного обучения на разнообразных данных.

Для устранения этих недостатков был предложен бенчмарк **SciRepEval** [23]. Он включает 24 разноформатные задачи, отражающие практические сценарии использования векторных представлений.

Центральной идеей бенчмарка является разделение задач на четыре концептуальных формата, основанное на гипотезе, что единое векторное представление не может быть оптимальным для всех типов задач одновременно:

- **Классификация (Classification, CLF)**: задачи, требующие отнесения документа к одной или нескольким категориям.
- **Регрессия (Regression, RGN)**: задачи, где необходимо предсказать непрерывную величину, связанную с документом (например, число цитирований).
- **Поиск по близости (Proximity, PRX)**: задачи, где запросом является один документ, и требуется найти другие, семантически близкие к нему документы.
- **Поиск по произвольному запросу (Ad-hoc Search, SRCH)**: классические задачи информационного поиска, где запрос представляет собой короткую текстовую строку.

Другим важным нововведением является разделение наборов данных на две группы: **In-Train** и **Out-of-Train**. Это разделение напрямую связано с подходом, предложенным в модели SPECTER2, где для одного документа предлагается

генерировать несколько векторных представлений, каждое из которых специализировано под свой формат задач. Для обучения таких формат-специфичных представлений необходимы соответствующие обучающие данные. Именно для этой цели в SciRepEval были собраны и выделены крупные наборы данных ‘In-Train’, которые авторы предоставили в открытом доступе как часть бенчмарка. Задачи из группы ‘Out-of-Train’ являются полностью отложенными и используются исключительно для оценки способности моделей, обученных на ‘In-Train’ задачах, обобщаться на новые данные и домены.

Для оценки качества в рамках SciRepEval используется широкий спектр мер, соответствующих каждому формату задач. Для классификации применяются бинарная и макро-усредненная **F1-мера**. Для задач регрессии используется ранговая **корреляция Кендалла (τ)**, устойчивая к выбросам и нелинейным зависимостям. Качество решения задач поиска оценивается с помощью стандартных мер ранжирования, таких как **MAP** и **nDCG**.

1.7.3 Бенчмарк SciFact

В то время как бенчмарки SciDocs и SciRepEval сосредоточены на оценке семантической близости и решении задач классификации или ранжирования, существует качественно иной тип задач, требующий от моделей не простого сопоставления текстов, а логического вывода и верификации фактов. Для оценки моделей в этом, более сложном, сценарии был разработан бенчмарк **SciFact** [28].

Задача, которую ставит SciFact, — это **верификация научного утверждения**. Модели необходимо для заданного утверждения (claim), представляющего собой атомарное, проверяемое высказывание, найти в корпусе научных статей релевантные аннотации и определить, **подтверждает (SUPPORTS)** или **опровергает (REFUTES)** текст аннотации данное утверждение.

Таким образом, задача верификации в SciFact является двухэтапной и включает в себя:

1. **Поиск доказательств (Evidence Retrieval)**. На первом этапе для заданного утверждения c модель должна произвести поиск по всему корпусу аннотаций \mathcal{A} и найти небольшое подмножество $\mathcal{E}(c) \subset \mathcal{A}$, содержащее релевантные доказательства.

2. **Предсказание отношения (Stance Prediction).** На втором этапе для каждой найденной пары (утверждение c , аннотация $a \in \mathcal{E}(c)$) модель должна вынести вердикт, классифицировав их отношение как ‘SUPPORTS’ или ‘REFUTES’.

Формальная постановка задачи. В обобщенном виде задачу можно сформулировать следующим образом.

Дано: Утверждение c и корпус аннотаций $\mathcal{A} = \{a_i\}_{i=1}^N$.

Найти: Множество предсказаний $\mathcal{S}_{\text{pred}} = \{(a_j, l_j, R_j)\}$,

где для каждого предсказания j :

- $a_j \in \mathcal{A}$ — аннотация, найденная как содержащая доказательство.
- $l_j \in \{\text{SUPPORTS}, \text{REFUTES}\}$ — предсказанное отношение между c и a_j .
- $R_j \subseteq \text{sentences}(a_j)$ — предсказанный рациональ, т.е. подмножество предложений из a_j , обосновывающее метку l_j .

Ключевой особенностью и усложняющим фактором бенчмарка SciFact является введение понятия **рационаля (rationale)**. Для каждой пары (утверждение, аннотация) с меткой ‘SUPPORTS’ или ‘REFUTES’ в наборе данных также размечен «рациональ» — минимальный набор предложений из аннотации, который является достаточным для эксперта, чтобы обосновать вынесенный вердикт. Это требование подталкивает к созданию интерпретируемых моделей, способных не просто классифицировать, но и указывать на конкретные фрагменты текста, послужившие основанием для вывода.

Оценка качества моделей на бенчмарке SciFact производится на двух уровнях детализации, и в качестве основной меры качества на обоих уровнях используется **F1-мера**.

- **Уровень аннотаций (Abstract Level).** Оценивается способность системы в целом правильно находить аннотации с доказательствами и корректно их классифицировать. Предсказание считается верным, если модель нашла релевантную аннотацию и правильно определила ее отношение к утверждению (‘SUPPORTS’ или ‘REFUTES’).
- **Уровень предложений (Sentence Level).** Более строгая оценка, которая дополнительно проверяет, смогла ли модель правильно выделить рациональ. Предсказание засчитывается, только если помимо правильного поиска и классификации аннотации модель также корректно определила набор предложений, составляющих рациональ.

SciFact представляет собой важный шаг в развитии методов оценки, смещая фокус с задач семантического сопоставления на задачи, требующие от моделей логического анализа, поиска доказательств и интерпретируемости результатов.

1.7.4 Универсальные бенчмарки: МТЕВ

Проблему фрагментированной оценки, когда модели тестируются лишь на нескольких специфичных задачах (например, только на поиске или только на определении семантической близости), решают универсальные наборы для оценки. Ключевым из них на сегодняшний день является **МТЕВ** (Massive Text Embedding Benchmark) [25].

МТЕВ не вводит принципиально новых задач, а объединяет и стандартизирует большое число уже известных и проверенных научным сообществом наборов данных, включая многие из рассмотренных ранее. Он охватывает широкий спектр из 8 типов задач.

Основной вклад МТЕВ заключается в предоставлении единого программного интерфейса (API) и открытой библиотеки, которая позволяет с минимальными усилиями оценить любую модель векторного представления на всем многообразии задач. Результаты всех моделей собираются в публичную таблицу лидеров, которая служит общепринятым стандартом для сравнения и выбора оптимальной модели для конкретного прикладного сценария.

1.8 Основные выводы

Современные методы построения семантических векторных представлений в значительной степени опираются на архитектуру трансформера и двухэтапную схему обучения. На первом этапе модель проходит предобучение без учителя на больших текстовых корпусах, а на втором дообучается с использованием контрастивных постановок и сиамских архитектур. Именно этот этап позволяет получить векторное пространство, в котором геометрическая близость отражает семантическую близость текстов. В специализированных предметных областях,

включая научный дискурс, особое значение приобретает доменная адаптация. Она может использовать характерные для научной коммуникации сигналы, в том числе граф цитирований, как это реализовано в моделях SPECTER и SciNCL.

В то же время существующие решения имеют существенные ограничения. Большая часть рассмотренных специализированных моделей и бенчмарков, включая SciDocs и SciRepEval, ориентирована на английский язык. Это затрудняет их применение к научным корпусам на других языках, в том числе на русском. Кроме того, недостаточно разработаны подходы к построению двуязычных семантических пространств для научных текстов, необходимых для кросс-языкового поиска.

Указанные ограничения определяют научную задачу настоящей диссертационной работы: разработку и исследование методов построения эффективных двуязычных семантических векторных представлений для научных документов.

Глава 2. Разработка и обучение двуязычных моделей векторизации научных текстов SciRus

В настоящей главе описывается процесс разработки семейства легковесных двуязычных моделей SciRus, предназначенных для построения семантических векторных представлений научных текстов. Описывается общая постановка задачи, излагается и обосновывается выбранная архитектура моделей. Детально рассматривается двухэтапная методология обучения, включающая предобучение на большом корпусе неразмеченных научных статей и последующее контрастивное дообучение для формирования семантически согласованного кросс-язычного векторного пространства. Приводятся сведения об использованных наборах данных и параметрах обучения.

2.1 Постановка задачи и существующие ограничения

Задачей, решаемой в данной работе, является построение семантических векторных представлений для коллекции научных документов. Формально, дана коллекция документов $\mathcal{D} = \{x_i\}_{i=1}^N$, где каждый документ x_i представляет собой последовательность токенов. Необходимо найти параметризованное отображение $f(\cdot, \alpha)$, которое сопоставляет каждому документу x_i вектор \mathbf{v}_i в пространстве \mathbb{R}^d :

$$\mathbf{v}_i = f(x_i, \alpha) \in \mathbb{R}^d.$$

Ключевое требование к этому отображению заключается в том, чтобы геометрическая близость векторов \mathbf{v}_i и \mathbf{v}_j в пространстве \mathbb{R}^d соответствовала семантической близости исходных документов x_i и x_j .

Современная парадигма решения этой задачи, как следует из обзора литературы (Глава 1), опирается на архитектуру трансформер-кодировщика и предполагает двухэтапный процесс обучения, в ходе которого параметры α оптимизируются с помощью градиентного спуска. На первом этапе модель проходит предобучение без учителя на больших неразмеченных текстовых корпусах. Цель этого этапа — выучивание моделью общих языковых закономерностей, синтаксиса и семантики. В качестве критерия используется, как правило, задача

маскированного языкового моделирования (MLM):

$$\sum_{i=1}^N \mathcal{L}_{\text{MLM},i}(\alpha) \rightarrow \min_{\alpha}.$$

Однако, как было показано в разделе 1.5, векторные представления, полученные непосредственно после этапа предобучения, показывают низкое качество на задачах семантического поиска. Для формирования векторного пространства требуется второй этап — контрастивное дообучение. На этом этапе модель обучается на парах или тройках текстов с известными семантическими отношениями с целью сблизить векторы семантически схожих документов и отдалить векторы семантически различающихся документов:

$$\sum_{i=1}^N \mathcal{L}_{\text{Contr},i}(\alpha) \rightarrow \min_{\alpha}.$$

После обучения кодировщик $f(\cdot, \alpha)$ может использоваться как основа для решения разных прикладных задач. Его параметры α фиксируются, а для новой задачи, например классификации или регрессии, обучается только легковесная модель $g'(\cdot, \beta')$ на сравнительно небольшом наборе данных $\mathcal{D}' = \{x'_i\}_{i=1}^M$, где $M \ll N$:

$$\sum_{i=1}^M \mathcal{L}'_i(g'(f(x'_i, \alpha), \beta')) \rightarrow \min_{\beta'}, \quad (2.1)$$

при этом число параметров новой модели значительно меньше числа параметров кодировщика: $\dim(\beta') \ll \dim(\alpha)$. Такой подход позволяет переносить знания, полученные на этапе обучения кодировщика, и снижает вычислительные затраты при адаптации к новым задачам.

Несмотря на значительные успехи в этой области, анализ существующих решений (разделы 1.6, 1.7) выявил два ключевых ограничения. Первое — доменная и языковая ограниченность. Подавляющее большинство передовых моделей (SPECTER [22], SciNCL [24]) разработано и ориентировано исключительно на англоязычный научный корпус. Это создает существенный дефицит эффективных инструментов для анализа и поиска в массивах русскоязычных научных публикаций. Второе ограничение связано с недостатком эффективных кросс-языковых моделей для научной области. Для поиска между русскими и английскими научными текстами необходимо единое семантическое пространство, в котором

близкие по смыслу документы располагаются рядом независимо от языка публикации. Такое пространство также важно для сопоставления исследований, опубликованных на разных языках.

Для преодоления указанных ограничений в рамках данной работы были разработаны модели SciRus. Основными целями при их создании являлись:

- Вычислительная эффективность и легковесность. Создание моделей с малым числом параметров, допускающих обучение и применение при ограниченных вычислительных ресурсах.
- Двуязычность. Обеспечение высокого качества векторных представлений как для русского, так и для английского языков, и формирование единого векторного пространства для эффективного кросс-языкового поиска.
- Высокое качество представлений при малом числе параметров. Разработанные модели должны демонстрировать более высокие показатели качества на целевых задачах по сравнению с аналогами сопоставимого размера. Вместе с тем, их качество должно быть конкурентоспособным на фоне значительно более крупных и ресурсоемких моделей.

Полный спектр целевых задач, используемых для всесторонней оценки качества разработанных моделей, будет подробно представлен в последующих главах, посвященных бенчмаркам RuSciBench (Глава 3) и RuSciFact (Глава 4).

Далее в главе подробно описывается процесс разработки, обучения и оценки моделей SciRus, направленный на достижение поставленных целей.

2.2 Наборы данных для обучения

Для обучения моделей SciRus были использованы два крупных мультиязычных корпуса научных текстов: Semantic Scholar Open Research Corpus (S2ORC) и данные российской научной электронной библиотеки eLibrary.ru.

Semantic Scholar Open Research Corpus (S2ORC) [60] представляет собой обширный гетерогенный граф знаний, агрегирующий информацию о научных публикациях, авторах и цитированиях из различных источников, включая Crossref, PubMed, Unpaywall и др. Исходно датасет содержит метаданные более чем 200 миллионов публикаций. Для обучения моделей SciRus из S2ORC была сформирована выборка, содержащая заголовки и аннотации научных статей.

Общий объем этой выборки составил около 30,5 миллионов пар ”заголовок-аннотация”.

В дополнение к текстовым метаданным, ключевым ресурсом для обучения является граф цитирований, предоставляемый S2ORC. Полный граф является чрезвычайно разреженным и содержит более 2.5 миллиардов ребер. Ввиду его масштаба, для обучения была использована подвыборка. Для ее формирования была произведена случайная выборка вершин (статей), после чего в итоговый граф были включены только те ребра, которые соединяют вершины из этой выборки. Полученный таким образом подграф содержит 51 970 696 ребер (цитирований). Этот сигнал о цитировании, как было показано в работах [22; 24], является семантически богатым источником для формирования положительных пар в задачах контрастивного обучения.

Выборка из S2ORC является мультиязычной, хотя в ней преобладают англоязычные тексты, составляющие приблизительно 83.3%. Помимо английского языка, в корпус входят публикации на китайском (2.8%), французском (2.6%), испанском (2.4%), немецком (1.1%) и других языках. Доля русскоязычных текстов составляет около 0.4%. Наличие документов на разных языках, а также пар с различающимися языками заголовка и аннотации, создает условия для формирования многоязычных и кросс-языковых свойств моделей.

В англоязычной части выборки медианная длина заголовка составляет 81 символ (11 слов), а 95-й перцентиль равен 154 символам (21 слово). Для аннотаций медианная длина составляет 972 символа (144 слова), а 95-й перцентиль достигает 2388 символов (354 слова). Эти значения использовались при выборе максимальной длины входной последовательности моделей.

По тематическому составу выборка S2ORC охватывает широкий круг научных областей, включая медицину (14.5%), биологию (9.3%), физику (5.3%), инженерные науки (4.7%), компьютерные науки (4.4%), химию (4.2%) и другие направления. Такой состав корпуса способствует обучению представлений, пригодных для работы с текстами из разных научных областей.

Данные научной электронной библиотеки eLibrary.ru. Для усиления способностей моделей работать с русскоязычным научным контентом и улучшения кросс-языковых возможностей между русским и английским языками, в обучающий корпус были добавлены данные из крупнейшей российской информационно-аналитической системы eLibrary.ru. Из этого источника было извлечено около 17,7 миллионов пар ”заголовок-аннотация”.

Особенностью данных из eLibrary.ru является их сбалансированность по русскому и английскому языкам: примерно 8.6 миллионов документов представлено на русском языке и около 8.8 миллионов - на английском. Существенная часть русскоязычных статей (около 5.2 миллионов, согласно оценкам в [9]) имеет также англоязычные версии заголовка и аннотации, что фактически формирует параллельный корпус и является ценным ресурсом для обучения кросс-языковых моделей. Другие языки также присутствуют, но в значительно меньшем объеме (например, украинский - 0.75%, китайский - 0.56%).

Анализ длин текстов показывает, что русскоязычные заголовки имеют медианную длину 80 символов (9 слов), а англоязычные - 91 символ (12 слов). Русскоязычные аннотации в медиане содержат 441 символ (50 слов), англоязычные - 946 символов (138 слов).

Данные eLibrary.ru также содержат информацию о цитированиях. Построенный на их основе граф цитирований включает 39,988,291 ребро. Этот граф отражает связи преимущественно в русскоязычном и смешанном русско-английском научном пространстве и используется как второй источник структурной информации для формирования обучающих примеров.

Статьи из eLibrary.ru охватывают разные научные направления, в том числе физические науки (12.1%), клиническую медицину (10.0%), экономику и бизнес (9.7%), химические науки (8.6%), биологические науки (8.5%) и другие области.

Итоговый обучающий корпус был получен объединением выборок из S2ORC и eLibrary.ru. Его объем составил около 48.2 миллионов пар "заголовок-аннотация", что соответствует примерно 15 миллиардам токенов. Объединенный корпус сочетает широкий международный охват S2ORC с большим массивом русскоязычных и русско-английских параллельных текстов из eLibrary.ru. На этом корпусе выполнялось двухэтапное обучение моделей SciRus.

Чтобы исключить пересечение обучающих и тестовых данных, из обучающей выборки были удалены все примеры, статьи из которых входят в тестовые наборы RuSciBench и SciDocs. Такая фильтрация снижает риск утечки тестовых данных и позволяет оценивать качество моделей на ранее не встречавшихся примерах.

2.3 Архитектура моделей SciRus

В качестве базовой архитектуры для моделей SciRus была выбрана модель типа RoBERTa [46], являющаяся усовершенствованной версией архитектуры BERT [18]. RoBERTa использует только энкодерную часть трансформера [6]. Для обеспечения вычислительной легковесности были выбраны конфигурации с относительно небольшим числом параметров.

Конфигурация моделей:

- **SciRus-tiny**: размер векторного представления - 312, общее количество параметров - приблизительно 23 миллиона.
- **SciRus-small**: размер векторного представления - 768, общее количество параметров - приблизительно 61 миллион.

Также модели состоят из 3 слоев, имеют 12 голов внимания. Обе модели могут обрабатывать входные последовательности длиной до 1024 токена. Для токенизации текста используется алгоритм Byte-Pair Encoding (BPE) [43] на уровне байтов (byte-level BPE). Размер словаря для обеих моделей составляет 50265 токенов.

Анализ распределения параметров в моделях SciRus выявляет, что существенная их доля сконцентрирована в слоях векторных представлений токенов (эмбеддингов). Так, для модели SciRus-tiny на слои эмбеддингов приходится около 69.5% всех параметров (16.0 млн из 23.0 млн), а для SciRus-small это значение составляет примерно 64.3% (39.4 млн из 61.2 млн). Данная особенность характерна и для других компактных моделей: например, в ‘cointegrated/rubert-tiny’ [61] доля параметров эмбеддингов достигает 79.6% (9.4 млн из 11.8 млн), а в ‘cointegrated/rubert-tiny2’ [11] — даже 91.8% (26.8 млн из 29.2 млн). В то же время, у более крупных моделей, таких как ‘allenai/specter’ [22], это соотношение иное: на слои эмбеддингов приходится лишь около 22.1% параметров (24.3 млн из 109.9 млн).

Сокращение доли параметров, приходящихся на эмбеддинги, в компактных моделях ограничено требованиями к качеству токенизации. Число параметров слоя эмбеддингов определяется произведением размера словаря на размерность вектора эмбеддинга ($V \times H$, где V — размер словаря, H — размерность эмбеддинга), поэтому наиболее прямой способ уменьшить этот слой состоит в сокращении словаря. Однако слишком малый словарь ухудшает разбиение текста на токены. Многие слова, особенно в языках с развитой морфологией, таких как русский,

начинают представляться большим числом коротких субсловных единиц или отдельных символов. Это увеличивает длину токенизированной последовательности, повышает вычислительную нагрузку и может затруднять моделирование семантических связей. Поэтому для сохранения приемлемого качества токенизации требуется достаточно большой словарь, даже если в компактных моделях это увеличивает относительную долю параметров слоя эмбеддингов.

Выбранные конфигурации существенно меньше стандартных базовых моделей, таких как BERT-base с 110 млн параметров и RoBERTa-base со 125 млн параметров. Это снижает потребление памяти и повышает скорость применения модели. SciRus-tiny является наиболее легковесным вариантом, а SciRus-small представляет компромисс между вычислительной эффективностью и более высоким потенциальным качеством представлений за счет большего числа параметров. Максимальная длина входной последовательности увеличена до 1024 токенов по сравнению с типичными 512 токенами во многих моделях BERT-типа, что позволяет обрабатывать более длинные фрагменты, включая развернутые аннотации и другие научные тексты.

2.4 Методология обучения

Процесс обучения моделей SciRus состоял из двух последовательных этапов: предобучение с использованием задачи маскированного языкового моделирования (MLM) и последующее контрастивное дообучение.

2.4.1 Предобучение с использованием Masked Language Modeling (MLM)

На первом этапе модели SciRus проходили предобучение с нуля, инициализируясь случайными весами. В качестве обучающих данных на этом этапе использовался исключительно объединенный текстовый корпус, состоящий из заголовков и аннотаций из наборов данных S2ORC и eLibrary.ru. Структурная информация из графов цитирований на данном этапе не применялась.

Обучение проводилось с использованием задачи маскированного языкового моделирования (MLM), унаследованной от архитектуры RoBERTa, с применением динамического маскирования. Формальное описание данного метода и параметры маскирования были подробно изложены в разделе 1.4. Оптимизация параметров моделей осуществлялась путем минимизации функции потерь перекрестной энтропии, формальное определение которой было также дано в обзоре литературы (раздел 1.4).

Процесс предобучения продолжался в течение двух эпох. Для контроля сходимости и предотвращения переобучения была сформирована валидационная подвыборка размером 1% от общего объема данных. График сходимости функции потерь (Рисунок 2.1) демонстрирует синхронное снижение ошибки как на обучающей, так и на валидационной выборках, с выходом на плато к концу второй эпохи. Анализ промежуточных результатов на бенчмарках SciDocs и RuSciBench (Рисунки 2.2 и 2.3) также подтвердил, что дальнейшее обучение не приводило к значимому улучшению качества представлений.

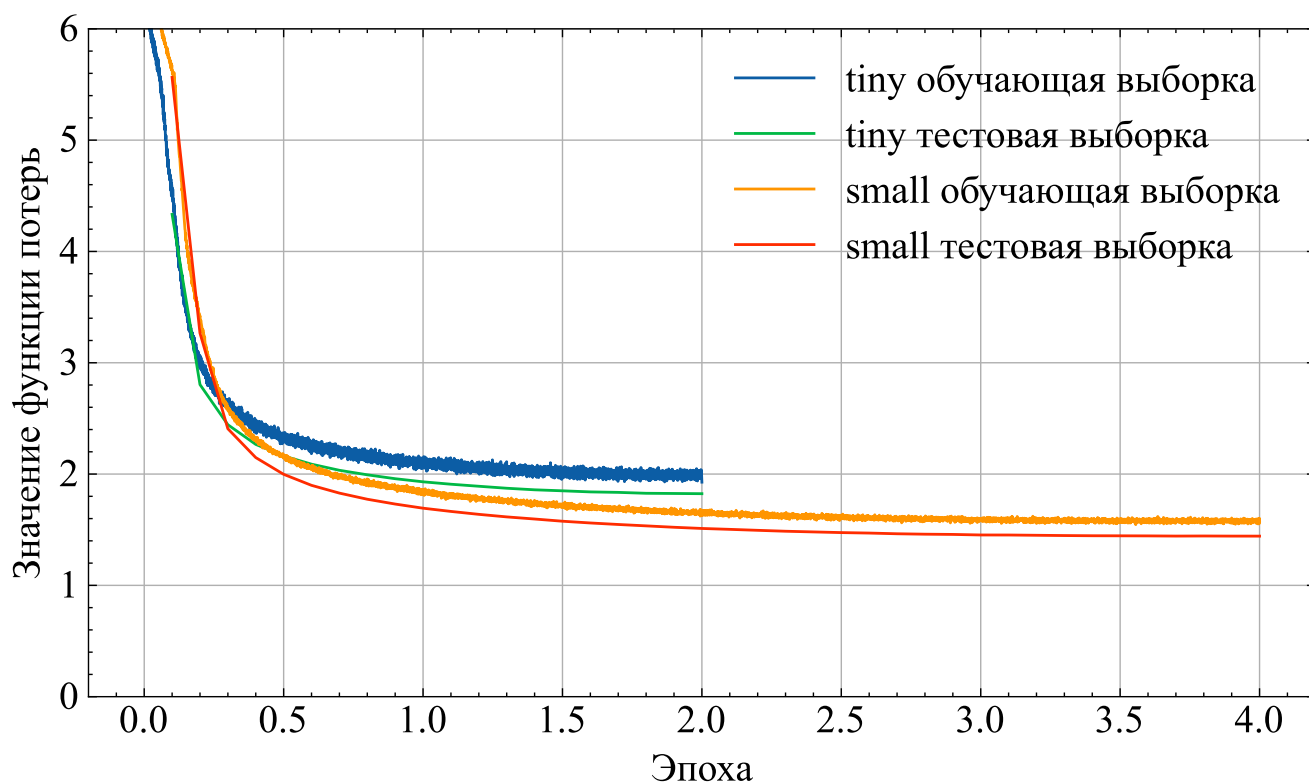


Рисунок 2.1 — Значение функции ошибки моделей SciRus

После завершения этапа MLM-предобучения модели приобретают способность генерировать контекстуализированные векторные представления для отдельных токенов, а не для текстов как целостных единиц. Эти представления, полученные на уровне токенов, еще не оптимизированы для задач семантического

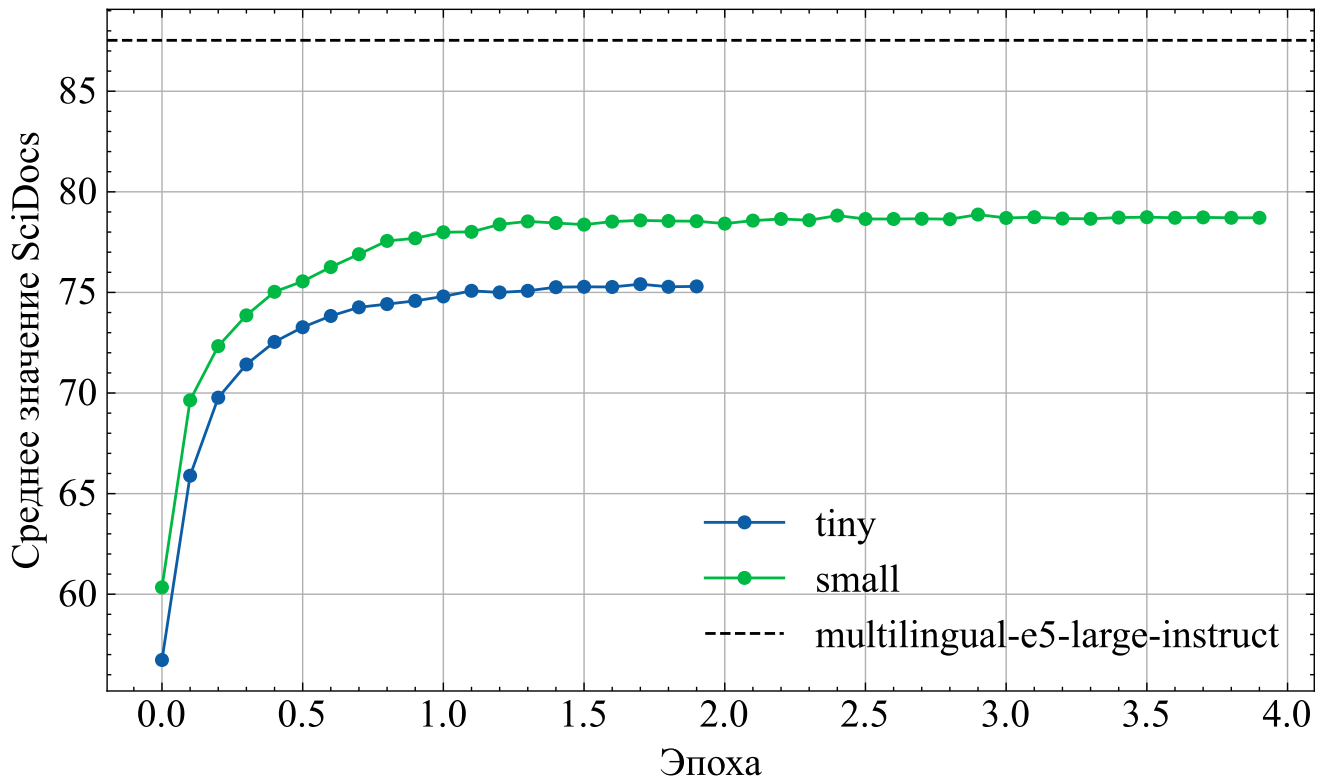


Рисунок 2.2 — Прогресс обучения моделей SciRus на англоязычном бенчмарке SciDocs.

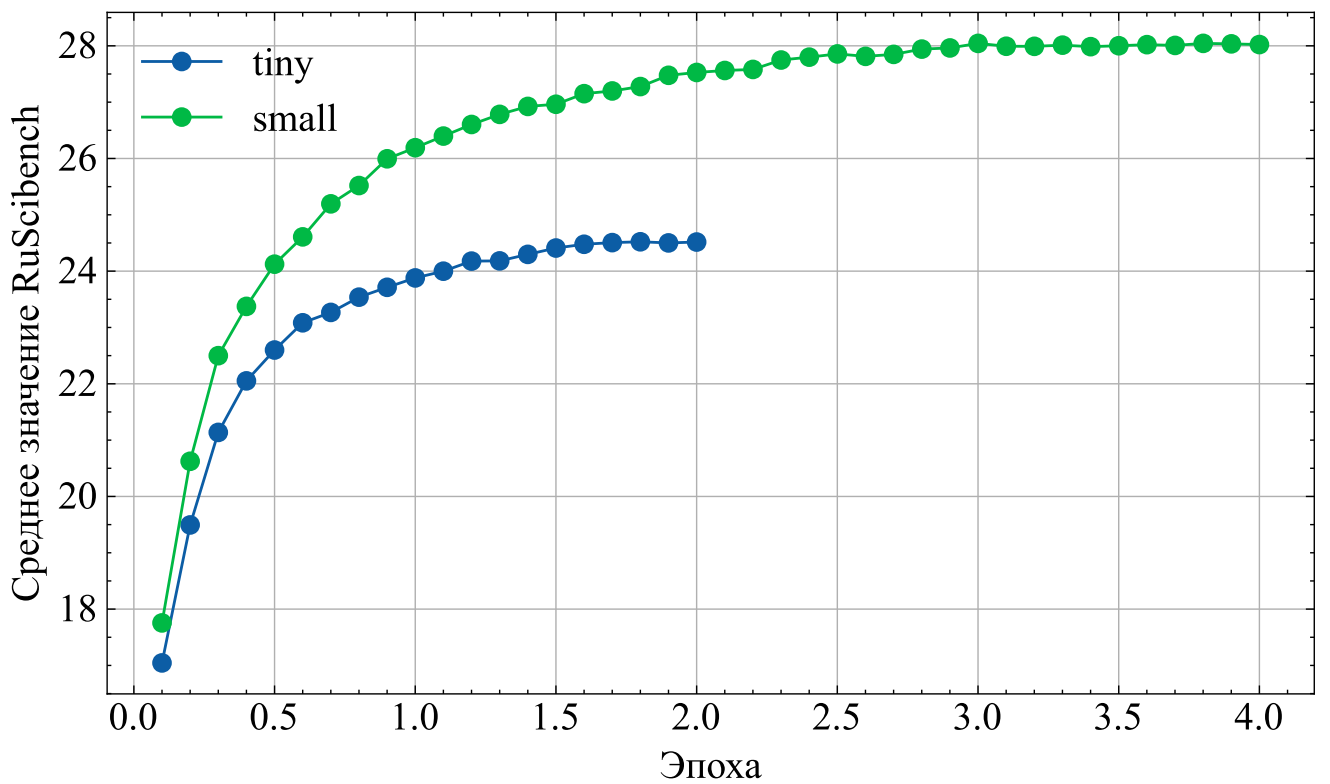


Рисунок 2.3 — Прогресс обучения моделей SciRus на русскоязычном бенчмарке RuSciBench.

поиска или сравнения текстов на уровне документа. Для этой цели предназначен второй этап обучения.

2.4.2 Контрастивное дообучение

На втором этапе модели, предобученные с помощью MLM, проходили дообучение с использованием контрастивного подхода. Цель этого этапа — преобразовать векторные представления, полученные на уровне токенов, в семантически согласованные векторы на уровне документов, пригодные для задач поиска и сравнения. Для агрегации векторов токенов в единый вектор документа применялся метод усреднения (Mean Pooling), как и в ряде других успешных моделей [8; 19].

В качестве основного метода обучения использовалась функция потерь InfoNCE, формальное описание которой приведено в разделе 1.5 (см. формулу 1.3). В качестве меры близости $s(\cdot, \cdot)$ применялась косинусная близость, а температурный коэффициент τ был установлен равным 0.01. Во всех случаях отрицательные примеры для каждой опорной статьи формировались из других примеров в той же мини-выборке (стратегия "in-batch negatives").

В рамках исследования были реализованы и обучены две версии моделей, различающиеся наборами данных и, соответственно, типами семантических сигналов, использованных для контрастивного обучения.

Обучение на парах «заголовок–аннотация». Первая версия моделей обучалась исключительно на парах, сформированных из текстовых метаданных. В этом подходе для формирования i -й положительной пары $(x_{a,i}, x_{p,i})$ в качестве опорного примера (anchor) $x_{a,i}$ использовался заголовок научной статьи, а в качестве положительного примера (positive) $x_{p,i}$ — аннотация той же статьи. Этот метод основан на предположении о высокой семантической близости между заголовком и аннотацией одного документа.

Ключевой особенностью этого подхода является обучение кросс-языковых представлений. Для статей из набора данных eLibrary.ru, имеющих аннотации на русском и английском языках, пары формировались случайным образом. Например, в качестве опорного примера мог быть выбран заголовок на русском языке, а

в качестве положительного — аннотация на английском. Такая стратегия заставляет модель выравнивать векторные пространства для разных языков, обучаясь сопоставлять семантически эквивалентные тексты независимо от языка их написания.

Обучение с использованием комбинированных данных. Для обучения второй версии моделей был сформирован объединенный набор данных, представляющий собой конкатенацию двух типов положительных пар: (1) пар «заголовок–аннотация», описанных в предыдущем пункте, и (2) пар, сформированных на основе графов цитирований S2ORC и eLibrary.ru. При формировании пар на основе цитирований в качестве опорного примера $x_{a,i}$ использовалась конкатенация заголовка и аннотации цитирующей статьи, а в качестве положительного примера $x_{p,i}$ — конкатенация заголовка и аннотации цитируемой статьи. Кросс-языковая стратегия сэмплирования применялась для всего объединенного набора данных: опорный и положительный примеры могли быть представлены на разных языках, что дополнительно усиливало выравнивание семантических пространств.

В результате были получены две линейки моделей. Модели, обученные исключительно на парах «заголовок–аннотация», далее обозначаются как SciRus-tiny и SciRus-small. Модели, при обучении которых дополнительно использовались данные о цитированиях, обозначаются с суффиксом -cite: SciRus-tiny-cite и SciRus-small-cite.

Динамика улучшения качества на бенчмарках SciDocs и RuSciBench в ходе контрастивного дообучения представлена на Рисунках 2.4 и 2.5 соответственно. Обучение также проводилось в течение 2 эпох. Этот этап позволил моделям SciRus достичь высоких результатов, в том числе на задачах кросс-языкового поиска, что подтверждает эффективность выбранных стратегий обучения.

2.5 Оценка качества моделей SciRus

Для всесторонней оценки качества разработанных моделей SciRus был проведен их сравнительный анализ с другими известными моделями. Отбор моделей для сравнения производился по двум ключевым принципам. Во-первых, в состав

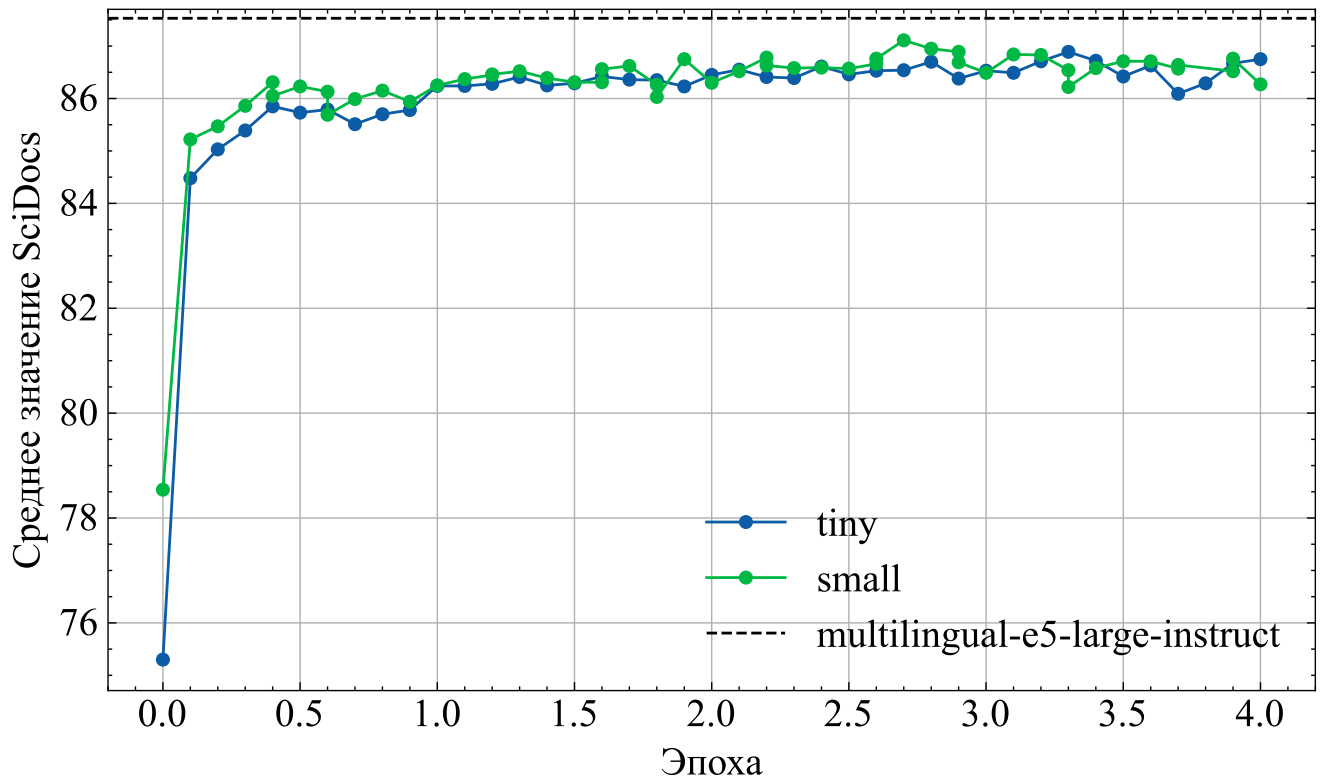


Рисунок 2.4 — Прогресс дообучения моделей SciRus на англоязычном бенчмарке SciDocs.

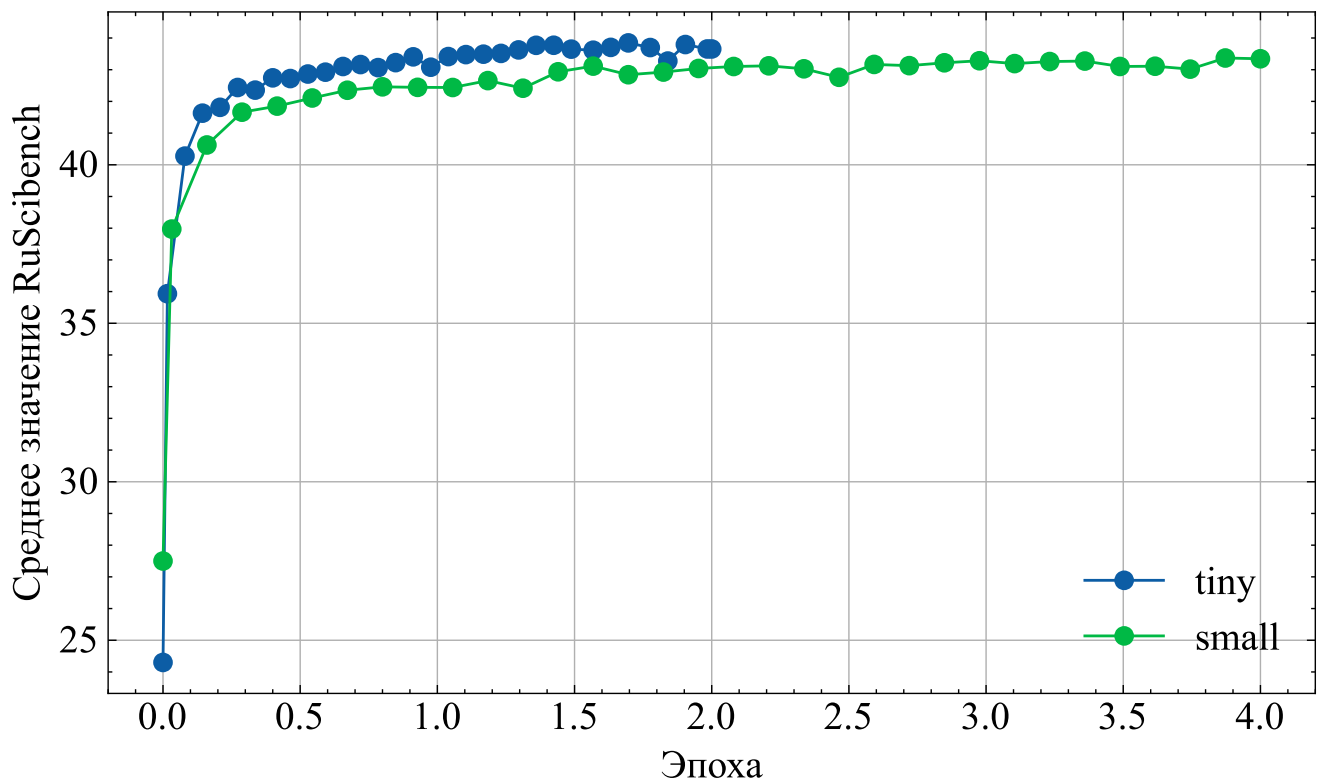


Рисунок 2.5 — Прогресс дообучения моделей SciRus на русскоязычном бенчмарке RuSciBench.

сопоставляемых решений были включены ведущие общецелевые англоязычные и многоязычные модели, занимающие лидирующие позиции в авторитетном международном лидерборде MTEB (Massive Text Embedding Benchmark) [25]. Такой выбор позволяет сопоставить модели SciRus с наиболее сильными современными решениями, не имеющими узкой доменной специализации. Во-вторых, для корректной оценки в рамках целевой области, в сравнение были добавлены специализированные модели, разработанные для векторизации именно научных текстов, такие как SPECTER [22] и SciNCL [24]. Такой двухкомпонентный подход к формированию выборки для сравнения обеспечивает возможность объективно оценить как конкурентоспособность моделей SciRus на фоне универсальных лидеров, так и их преимущества в решении задач научной области. Полный перечень использованных моделей, упорядоченный по дате публикации, представлен в таблице 1.

Таблица 1 — Список моделей для сравнения

| Модель | Количество параметров | Дата публикации | Ссылка |
|---|-----------------------|-----------------|---|
| SPECTER | 110 млн | 01-2021 | allenai/specter |
| SciNCL | 110 млн | 02-2022 | malteos/scincl |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 278 млн | 03-2022 | symanto/sn-xlm-roberta-base-snli-mnli-anli-xnli |
| paraphrase-multilingual-mpnet-base-v2 | 278 млн | 03-2022 | sentence-transformers/paraphrase-multilingual-mpnet-base-v2 |
| LaBSE-en-ru | 129 млн | 03-2022 | cointegrated/LaBSE-en-ru |
| multilingual-e5-base | 278 млн | 05-2023 | intfloat/multilingual-e5-base |
| multilingual-e5-large | 560 млн | 06-2023 | intfloat/multilingual-e5-large |
| multilingual-e5-small | 118 млн | 06-2023 | intfloat/multilingual-e5-small |

Продолжение таблицы 1

| Модель | Количество параметров | Дата публикации | Ссылка |
|--------------------------------|-----------------------|-----------------|---|
| SFR-Embedding-Mistral | 7 млрд | 01-2024 | Salesforce/SFR-Embedding-Mistral |
| GritLM-7B | 7.24 млрд | 02-2024 | GritLM/GritLM-7B |
| multilingual-e5-large-instruct | 560 млн | 02-2024 | intfloat/multilingual-e5-large-instruct |
| GIST-large-Embedding-v0 | 335 млн | 02-2024 | avsolatorio/GIST-large-Embedding-v0 |
| Linq-Embed-Mistral | 7 млрд | 05-2024 | Linq-AI-Research/Linq-Embed-Mistral |
| SciRus-small-cite | 61 млн | 05-2024 | mlsa-iai-msu-lab/sci-rus-small-cite |
| SciRus-tiny-cite | 23 млн | 05-2024 | mlsa-iai-msu-lab/sci-rus-tiny3-cite |
| SciRus-small | 61 млн | 05-2024 | mlsa-iai-msu-lab/sci-rus-small |
| SciRus-tiny | 23 млн | 05-2024 | mlsa-iai-msu-lab/sci-rus-tiny |

Для оценки и сопоставления с существующими решениями было проведено сравнительное тестирование разработанных моделей на общепринятом англоязычном бенчмарке SciDocs [22]. На момент проведения исследования устоявшиеся и публично доступные наборы данных для оценки качества семантических представлений в русскоязычной научной области отсутствовали. Именно для устранения этого пробела в рамках настоящей диссертационной работы были разработаны специализированные бенчмарки RuSciBench и RuSciFact. Детальное описание этих наборов данных, а также исчерпывающие результаты сравнительного тестирования всех моделей на них, будут представлены в последующих главах 3 и 4 соответственно. Результаты на англоязычном бенчмарке SciDocs приведены в таблице 2.

Анализ результатов, представленных в таблице 2, позволяет сделать следующие выводы. На англоязычном наборе данных SciDocs наилучшие

| Model | MAG | MeSH | Cite | | CoView | | CoCite | | CoRead | | Среднее |
|---|-------|-------|-------|-------|--------|-------|--------|-------|--------|-------|--------------|
| | | | map | nDCG | map | nDCG | map | nDCG | map | nDCG | |
| SciRus-tiny | 82.01 | 85.98 | 84.17 | 93.03 | 81.37 | 90.46 | 83.31 | 92.59 | 81.45 | 90.88 | 86.53 |
| SciRus-small | 81.88 | 88.28 | 83.66 | 92.90 | 81.29 | 90.53 | 84.07 | 92.96 | 82.13 | 91.15 | 86.89 |
| SciRus-tiny-cite | 82.01 | 89.48 | 90.30 | 95.83 | 84.19 | 91.76 | 88.85 | 95.18 | 85.21 | 92.64 | 89.55 |
| SciRus-small-cite | 83.13 | 90.25 | 89.80 | 95.50 | 84.64 | 91.99 | 89.91 | 95.65 | 86.15 | 93.20 | 90.02 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 75.05 | 72.97 | 68.41 | 84.33 | 70.06 | 84.22 | 67.71 | 83.60 | 68.94 | 83.66 | 75.89 |
| LaBSE-en-ru | 78.87 | 73.46 | 70.77 | 85.89 | 74.61 | 87.03 | 74.85 | 88.07 | 73.24 | 86.47 | 79.33 |
| multilingual-e5-small | 82.14 | 88.06 | 81.23 | 91.71 | 81.26 | 90.37 | 83.38 | 92.73 | 80.78 | 90.60 | 86.23 |
| multilingual-e5-base | 82.11 | 88.86 | 82.79 | 92.43 | 81.37 | 90.38 | 84.67 | 93.29 | 82.05 | 91.27 | 86.92 |
| multilingual-e5-large | 83.40 | 89.97 | 83.35 | 92.74 | 81.76 | 90.50 | 85.74 | 93.86 | 82.48 | 91.46 | 87.53 |
| Linq-Embed-Mistral | 79.63 | 88.43 | 86.94 | 94.42 | 84.08 | 91.87 | 88.12 | 94.97 | 85.97 | 93.27 | 88.77 |
| SFR-Embedding-Mistral | 79.05 | 87.70 | 86.67 | 94.35 | 84.92 | 92.26 | 88.44 | 95.07 | 86.64 | 93.60 | 88.87 |
| SPECTER | 79.40 | 87.70 | 92.00 | 96.60 | 83.40 | 91.40 | 88.00 | 94.70 | 85.10 | 92.70 | 89.10 |
| multilingual-e5-large-instruct | 83.49 | 89.67 | 86.87 | 94.49 | 84.20 | 91.88 | 88.17 | 94.94 | 85.22 | 92.88 | 89.18 |
| GritLM-7B | 84.63 | 90.38 | 88.19 | 95.04 | 84.12 | 91.72 | 88.92 | 95.22 | 85.68 | 93.12 | 89.70 |
| SciNCL | 81.11 | 89.00 | 93.55 | 97.35 | 85.28 | 92.23 | 91.66 | 96.44 | 87.69 | 94.00 | 90.84 |
| paraphrase-multilingual-mpnet-base-v2 | 82.61 | 89.52 | 92.97 | 97.04 | 85.67 | 92.44 | 92.37 | 96.79 | 87.18 | 93.74 | 91.03 |
| GIST-large-Embedding-v0 | 82.81 | 90.79 | 93.30 | 97.27 | 85.72 | 92.50 | 91.95 | 96.63 | 87.61 | 93.99 | 91.26 |

Таблица 2 — Сравнение моделей на бенчмарке SciDocs

средние показатели качества демонстрируют модели GIST-large-Embedding-v0, paraphrase-multilingual-mpnet-base-v2 и SciNCL.

Вместе с тем, разработанные модели семейства SciRus, обученные с использованием данных о цитированиях, показывают высокий и конкурентоспособный уровень. Так, модель SciRus-small-cite со средним значением 90.02 достигает качества, сопоставимого с результатами специализированной модели SciNCL, уступая ей менее одного процентного пункта, но при этом имея почти вдвое меньшее число параметров (61 млн против 110 млн). Более того, данная модель превосходит не только SPECTER, но и значительно более крупные модели общего назначения, такие как multilingual-e5-large-instruct и GritLM-7B. Наиболее легковесный вариант, SciRus-tiny-cite (23 млн параметров), также демонстрирует высокий результат.

Следует отметить, что распределение лидеров по отдельным задачам неоднородно. В задачах тематической классификации (MAG, MeSH) преимущество остается за крупными англоязычными кодировщиками, такими как GritLM и GIST-large-Embedding-v0. В задачах, основанных на анализе структуры цитирований, наилучшие результаты показывают SciNCL (прямое цитирование, Cite) и paraphrase-multilingual-mpnet-base-v2 (совместная цитируемость, Co-cite). Наконец, в задачах, связанных с пользовательской активностью, лидируют GIST (совместные просмотры, Co-view) и SciNCL (совместные прочтения, Co-read).

Таким образом, полученные результаты подтверждают, что разработанные легковесные модели SciRus, обученные с использованием информации о цитированиях, не только сокращают разрыв в качестве с передовыми и значительно более ресурсоемкими англоязычными аналогами, но и превосходят многие из них. Это утверждение справедливо в том числе и при сравнении со специализированными научными моделями.

Следует особо подчеркнуть, что ведущие модели для научной области, SPECTER и SciNCL, ограничены поддержкой исключительно английского языка. На их фоне модели SciRus не только демонстрируют сопоставимое, а в ряде случаев и более высокое качество при меньшем числе параметров, но и являются полностью двуязычными, что является их принципиальным преимуществом. Это подтверждает высокую эффективность выбранной стратегии обучения для формирования компактных, но при этом высококачественных и кросс-языковых семантических представлений научных текстов.

2.6 Оценка производительности

В дополнение к качественным характеристикам, одной из ключевых целей при разработке моделей SciRus являлось обеспечение их высокой вычислительной эффективности и легковесности. Для экспериментального подтверждения достижения этой цели были проведены замеры скорости векторизации текстов на центральном процессоре (CPU). Для применения моделей и проведения тестов использовался фреймворк Text Embeddings Inference [62]. Данное решение было выбрано, поскольку оно предназначено для промышленного применения моделей векторизации. Тестирование всех моделей проводилось в одинаковых условиях на CPU Intel(R) Xeon(R) CPU @ 2.20GHz с использованием 2 ядер. В качестве инструмента для нагрузочного тестирования использовалась библиотека k6 [63], представляющая собой современное средство для оценки производительности систем. В ходе эксперимента каждая сравниваемая модель обрабатывала запросы в течение 60 секунд. В качестве входных данных для моделей использовались случайно выбранные тексты, представляющие собой конкатенацию заголовка и аннотации научной статьи из набора данных, сформированного на основе пуб-

ликаций научной электронной библиотеки eLibrary.ru. Полученные данные о времени инференса на CPU для различных моделей сведены в таблицу 3.

Таблица 3 — Сравнение среднего времени инференса различных моделей на CPU.

| Модель | Количество параметров (млн.) | Среднее время инференса (с) | Время инференса p95 (с) |
|---------------------------------------|------------------------------|-----------------------------|-------------------------|
| SciRus-tiny | 23 | 0.23 | 0.49 |
| SciRus-small | 61 | 0.44 | 0.95 |
| paraphrase-multilingual-mpnet-base-v2 | 278 | 1.61 | 3.26 |
| multilingual-e5-large-instruct | 560 | 5.19 | 9.27 |
| SciNCL | 110 | 6.77 | 14.71 |
| SPECTER | 110 | 9.14 | 16.13 |

Результаты, приведенные в таблице 3, демонстрируют существенное превосходство моделей семейства SciRus по скорости инференса на CPU. Модель SciRus-tiny показывает наилучшее среднее время отклика. Модель SciRus-small, имея большее число параметров, также демонстрирует высокую производительность, оказываясь примерно в 3.7 раза быстрее ближайшей по этому показателю модели paraphrase-multilingual-mpnet-base-v2.

2.7 Практическая апробация и внедрение модели SciRus-tiny

Одним из ключевых результатов диссертационной работы является не только разработка и оценка моделей, но и подтверждение их практической значимости путем внедрения в реальные информационно-поисковые системы. Для этой цели, в сотрудничестве с крупнейшей российской научной электронной библиотекой eLibrary.ru, был разработан и внедрен новый функционал семантического поиска, получивший название «нейропоиск». В качестве основы для данного функционала была выбрана легковесная модель SciRus-tiny, поскольку она продемонстрировала оптимальное сочетание высокого качества векторных представлений и вычислительной эффективности, что является критически важным для применения в высоконагруженных промышленных системах.

Процесс работы системы «нейропоиск» можно формально разделить на два основных этапа: предварительное индексирование и обработка поискового

запроса в режиме реального времени. На первом этапе была проведена векторизация всего корпуса аннотаций, доступных на портале eLibrary.ru. Для каждой аннотации x_i из коллекции D_{EL} было получено её семантическое векторное представление v_i с помощью разработанной модели:

$$v_i = f(x_i, \alpha_{\text{tiny}})$$

где f — функция модели-кодировщика, а α_{tiny} — параметры обученной модели SciRus-tiny. Полученный массив векторов $\{v_i\}$ был загружен в специализированную векторную базу данных, предназначенную для быстрого поиска ближайших соседей. На втором этапе пользовательский запрос x_q , представленный аннотацией, фрагментом или полным текстом документа, также преобразуется в вектор $v_q = f(x_q, \alpha_{\text{tiny}})$. После применения категориальных фильтров, таких как год публикации и тип документа, система выполняет поиск по отфильтрованной части векторной базы данных. Найденные публикации ранжируются по убыванию косинусной близости между вектором запроса v_q и векторами документов v_i . Интерфейс данного режима представлен на Рисунке 2.6.

The image shows a web interface for a semantic search system. At the top, there is a header with a logo and the text 'ПОИСК БЛИЗКИХ ПО ТЕМАТИКЕ ПУБЛИКАЦИЙ'. Below this is a section titled 'ПАРАМЕТРЫ' (Parameters). It contains several input fields and dropdown menus:

- 'Поисковый запрос: ?' (Search query) with a text input field containing the placeholder 'Аннотация, фрагмент текста или полный текст документа'.
- 'Год публикации:' (Publication year) with a dropdown menu set to 'все годы' (all years).
- 'Дата размещения на eLibrary.ru:' (Date of posting) with a dropdown menu set to 'за все время' (for all time).
- 'База данных:' (Database) with a dropdown menu set to 'входящие в РИНЦ' (included in RINCC).
- 'Тип публикации:' (Publication type) with a dropdown menu set to 'все типы публикаций' (all publication types).
- 'Сортировка:' (Sorting) with a dropdown menu set to 'по тематической близости' (by thematic proximity).
- 'Порядок:' (Order) with a dropdown menu set to 'по убыванию' (in descending order).

 At the bottom right of the parameter section are two buttons: 'Очистить' (Clear) and 'Поиск' (Search).

Рисунок 2.6 — Интерфейс режима «нейропоиск» на портале eLibrary.ru, использующего модель SciRus-tiny.

Внедрение семантического поиска на основе SciRus-tiny расширяет возможности работы с научными публикациями по сравнению с традиционным поиском

по ключевым словам. В первую очередь этот режим полезен при подготовке обзоров литературы, поскольку позволяет находить релевантные работы даже в тех случаях, когда они не содержат точных терминов из запроса, но близки к нему по смыслу. Кроме того, «нейропоиск» может применяться для подбора экспертов при рецензировании рукописей и оценке грантовых заявок, для выявления научных организаций и коллективов, работающих в заданной предметной области, а также для предварительного анализа при патентном поиске. Таким образом, данный инструмент переносит акцент с лексического совпадения на семантическую близость и тем самым расширяет возможности анализа научной информации.

2.8 Основные выводы

В настоящей главе был детально описан процесс разработки семейства легковесных двуязычных моделей SciRus, предназначенных для построения семантических векторных представлений научных текстов. Была обоснована выбранная архитектура на основе трансформера-кодировщика с уменьшенным числом параметров для обеспечения вычислительной эффективности и представлена двухэтапная методология обучения. На первом этапе, посредством задачи маскированного языкового моделирования на объединенном корпусе русско- и англоязычных статей, модели освоили общие языковые закономерности. На втором этапе, с помощью контрастивного дообучения, было сформировано единое семантическое пространство. Для этого использовались два типа обучающих сигналов: семантическая связь между заголовком и аннотацией, а также структурная информация из графов цитирований.

Проведенная предварительная оценка на англоязычном бенчмарке SciDocs продемонстрировала высокую конкурентоспособность разработанных моделей. В частности, было показано, что легковесные двуязычные модели SciRus достигают качества, сопоставимого и даже превосходящего качество значительно более крупных и исключительно англоязычных специализированных аналогов. Эксперименты по оценке производительности подтвердили их вычислительную эффективность, что было одной из ключевых целей разработки. Практическая значимость работы подтверждена успешным внедрением модели SciRus-tiny в промышленную эксплуатацию в составе информационно-поисковой системы на-

учной электронной библиотеки eLibrary.ru. Таким образом, в настоящей главе были созданы и верифицированы эффективные инструменты для семантического анализа научных текстов. Их всесторонняя и углубленная оценка на задачах русскоязычной и кросс-языковой научной области будет предметом рассмотрения в последующих главах, посвященных бенчмаркам RuSciBench и RuSciFact.

Глава 3. Мультизадачный бенчмарк для оценки моделей векторного представления русско- и англоязычных научных текстов

Объективная оценка и направленное совершенствование моделей векторизации текста невозможны без стандартизированных инструментов — бенчмарков. Такие инструменты объединяют наборы данных и задачи, позволяют проводить воспроизводимые эксперименты и получать сопоставимые результаты, необходимые для развития технологий обработки естественного языка. В последние годы для русского языка был разработан ряд общезыковых бенчмарков, включая RuSentEval [10] и encodechka [11]. Однако эти наборы не предназначены для систематической оценки моделей на научных текстах.

Научный дискурс имеет выраженную специфику. Для него характерны узкоспециализированная терминология, высокая информационная плотность, сложная синтаксическая структура и формальный стиль изложения. Поэтому качество модели, измеренное на общелитературных или новостных текстах, не позволяет надежно судить о ее эффективности в научном домене. Для английского языка существуют специализированные научные бенчмарки, например SciDocs [22]. Для русского языка такие инструменты до недавнего времени были представлены существенно слабее. Единственным известным примером является RuMedBench [64], но он относится только к медицинской области и не охватывает другие научные дисциплины.

Недостаток инструментов для оценки моделей на русскоязычном научном материале затрудняет сравнительный анализ существующих и вновь создаваемых моделей. Он также осложняет выбор подходящего решения для прикладных задач, включая семантический поиск и классификацию научных статей.

Для устранения этого пробела был разработан RuSciBench — мультизадачный и двуязычный бенчмарк, целенаправленно созданный для всесторонней оценки качества векторных представлений научных текстов на русском и английском языках. Он охватывает широкий спектр задач и предоставляет исследователям стандартизированную и воспроизводимую методологию тестирования. В настоящей главе будет подробно рассмотрена архитектура бенчмарка RuSciBench, описан состав входящих в него наборов данных и задач, представлена методология проведения оценки и приведены результаты, полученные для набора базовых моделей.

3.1 Источники данных и подготовка корпуса

В качестве основного источника данных для создания бенчмарка RuSciBench был выбран крупнейший российский ресурс научных публикаций – электронная библиотека eLibrary.ru¹. Эта платформа содержит обширный архив статей, диссертаций, монографий и материалов конференций, преимущественно на русском языке, с аннотациями на английском. Выбор eLibrary.ru обусловлен её представительностью для русскоязычной научной среды, а также наличием структурированных метаданных, включая рубрикацию, информацию о цитированиях и типах публикаций, что позволяет формировать разнообразные задачи оценки.

Для обеспечения двуязычности корпуса и возможности включения задач кросс-языкового поиска были отобраны только те статьи, для которых доступны аннотации как на русском, так и на английском языке. Это гарантирует сопоставимость представлений текстов на разных языках и позволяет оценивать модели в условиях, приближенных к реальным сценариям многоязычного анализа научных документов.

В процессе анализа данных были выявлены наиболее частотные проблемы с качеством данных, для устранения которых был разработан многоэтапный процесс очистки, включающий удаление всех HTML-тегов и специальных символов с использованием регулярных выражений, дедупликацию, фильтрацию аннотаций по минимальной длине для обеспечения достаточной информативности (более 50 символов), а также автоматическое определение языка текста. Определение языка проводилось с помощью библиотеки Lingua [65], выбранной из-за её высокой точности на коротких текстах (средняя точность для русского языка составляет 97.57%, для английского - 98.7% на наборе данных, состоящем из предложений) и превосходства над альтернативами (такими как langdetect[66] или fastText[67]) в сценариях со смешанными языками и шумными данными. Эта процедура была необходима, поскольку исходные метки языка в данных eLibrary.ru не всегда соответствовали реальности, что могло привести к искажениям в оценке моделей.

В результате применения процесса очистки было отфильтровано около 7,6% исходных записей. Финальный корпус содержит 182 264 статьи, каждая из

¹<https://www.elibrary.ru>

которых включает заголовок и аннотацию на русском и английском языках, а также метаданные. Статистические характеристики корпуса приведены в таблице 4, где указано распределение длин текстов.

Таблица 4 — Распределение числа символов в заголовках и аннотациях

| Язык | Тип | Среднее | 25% | 50% | 75% |
|------------|-----------|---------|-----|-----|------|
| Русский | Заголовок | 89 | 65 | 85 | 108 |
| Английский | Заголовок | 90 | 65 | 85 | 109 |
| Русский | Аннотация | 769 | 340 | 564 | 1044 |
| Английский | Аннотация | 807 | 347 | 586 | 1110 |

Подготовленный корпус обеспечивает высокое качество данных и позволяет проводить объективную оценку моделей векторных представлений в научном домене, учитывая специфику русскоязычных текстов.

3.2 Состав и методология оценки в бенчмарке RuSciBench

Бенчмарк RuSciBench, разработанный в соавторстве [36], представляет собой комплексный инструментарий, предназначенный для разносторонней оценки моделей семантического векторного представления научных текстов. Он включает задачи трех типов, отражающих ключевые сценарии использования векторных представлений в научной среде: классификация документов, регрессионный анализ и информационный поиск. Каждая из 9 уникальных задач представлена в двуязычном формате (на русском и английском языках), что в совокупности составляет 18 наборов данных. Такое разнообразие позволяет выявлять сильные и слабые стороны моделей, оценивая их способность кодировать как тематическую принадлежность, так и более тонкие семантические и структурные характеристики текста.

В рамках данной совместной работы вклад соискателя заключался в разработке и реализации задач, относящихся к каждому из этих трех типов. Соответственно, в последующих разделах детально описаны именно те составные части бенчмарка, которые были разработаны лично соискателем.

Для проведения всесторонней и объективной оценки на задачах бенчмарка RuSciBench был сформирован широкий и репрезентативный набор моделей-

конкурентов. Чтобы составить полную картину качества современных решений, отбор моделей производился на основе их позиций в двух авторитетных лидербордах, входящих в состав Massive Text Embedding Benchmark (МТЕВ) [25]: основного англоязычного (МТЕВ leaderboard) и его русскоязычной версии.

Важным аспектом является поддержание бенчмарка в актуальном состоянии, поскольку ценность такого инструмента определяется не только первоначальной публикацией, но и его способностью отражать текущее состояние области. В связи с этим, первоначальный список моделей, оцененных на момент публикации бенчмарка, был существенно расширен. В него были добавлены новые передовые модели, появившиеся после публикации. Таким образом, итоговый набор для оценки включает в себя все модели, представленные в таблице 1 из предыдущей главы, а также ряд актуальных на текущий момент решений. Перечень использованных для оценки моделей в дополнение к таблице 1, упорядоченный по дате их публикации, представлен в таблице 5.

Таблица 5 — Список моделей, оцененных на бенчмарке RuSciBench

| Модель | Количество параметров | Дата публикации | Ссылка |
|-------------------------|-----------------------|-----------------|---|
| rubert-tiny2 | 29 млн | 03-2022 | cointegrated/rubert-tiny2 |
| rubert-tiny | 12 млн | 03-2022 | cointegrated/rubert-tiny |
| SFR-Embedding-2_R | 7.11 млрд | 06-2024 | Salesforce/SFR-Embedding-2_R |
| gte-Qwen2-7B-instruct | 7 млрд | 06-2024 | Alibaba-NLP/gte-Qwen2-7B-instruct |
| gte-Qwen2-1.5B-instruct | 1 млрд | 06-2024 | Alibaba-NLP/gte-Qwen2-1.5B-instruct |
| USER-base | 124 млн | 06-2024 | deepvk/USER-base |
| rubert-tiny-turbo | 29 млн | 06-2024 | sergeyzh/rubert-tiny-turbo |
| USER-bge-m3 | 359 млн | 07-2024 | deepvk/USER-bge-m3 |
| NV-Embed-v2 | 7 млрд | 08-2024 | nvidia/NV-Embed-v2 |
| jina-embeddings-v3 | 572 млн | 09-2024 | jinaai/jina-embeddings-v3 |

Продолжение таблицы 5

| Модель | Количество параметров | Дата публикации | Ссылка |
|--------------------------|-----------------------|-----------------|--|
| Giga-Embeddings-instruct | 2 млрд | 12-2024 | ai-sage/Giga-Embeddings-instruct |
| FRIDA | 823 млн | 12-2024 | ai-forever/FRIDA |
| BERTA | 128 млн | 03-2025 | sergeyzh/BERTA |
| rubert-mini-frida | 32 млн | 03-2025 | sergeyzh/rubert-mini-frida |

3.2.1 Задачи классификации документов

Задачи классификации направлены на оценку способности векторного представления агрегировать и сохранять информацию о принадлежности документа к определенному семантическому классу. Формально, пусть дана коллекция документов $\mathcal{D} = \{x_i\}_{i=1}^N$ и конечное множество меток $C = \{c_1, \dots, c_K\}$, где K — число классов. Задача состоит в построении модели, способной сопоставить каждому документу x_i соответствующую метку $y_i \in C$.

Исходные данные для задач классификации характеризуются значительным дисбалансом классов. Чтобы обеспечить объективность оценки и избежать смещения в сторону мажоритарных классов, к обучающей и тестовой выборкам была применена процедура балансировки путем сокращения выборки (undersampling) примеров из преобладающих классов. Итоговый корпус данных для каждой задачи был разделен на обучающую и тестовую части в соотношении 90% к 10%.

В рамках бенчмарка используется подход «векторы как признаки» (embeddings as features). Это означает, что параметры модели-кодировщика $f(x, \alpha)$ заморожены, и для каждого документа x_i вычисляется его векторное представление $\mathbf{v}_i = f(x_i, \alpha)$. Затем на этих векторах обучается модель классификации - логистическая регрессия $g(\mathbf{v}, \theta)$.

Вероятность принадлежности документа x_i к классу c_k моделируется как:

$$p_{ik} = P(y_i = c_k | \mathbf{v}_i, \theta) = \frac{\exp(\mathbf{v}_i^T \theta_k)}{\sum_{j=1}^K \exp(\mathbf{v}_i^T \theta_j)},$$

где θ_k — вектор весов для класса c_k , а $\theta = \{\theta_1, \dots, \theta_K\}$ — полный набор параметров модели $g(\mathbf{v}, \theta)$

Процесс обучения сводится к минимизации функции потерь только по параметрам классификатора θ :

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K [y_i = c_k] \log p_{ik} + \lambda \|\theta\|_2^2 \longrightarrow \min_{\theta}, \quad (3.1)$$

где p_{ik} — вероятность принадлежности документа x_i к классу c_k , предсказанная моделью $g(\mathbf{v}_i, \theta)$, а второй член соответствует L2-регуляризации с коэффициентом λ . Для решения данной задачи оптимизации в экспериментах использовалась реализация логистической регрессии из библиотеки scikit-learn [68] со следующими параметрами: решатель lbfgs, L2-регуляризация с параметром $C=1.0$ (обратным коэффициенту регуляризации λ), мультиклассовая стратегия для многоклассовых задач и максимальное число итераций $\text{max_iter}=100$.

Поскольку тестовые выборки для задач классификации являются сбалансированными, в качестве основной меры качества используется точность (Accuracy). Она вычисляется как доля правильно классифицированных объектов и формально определяется следующим образом:

$$\text{Accuracy} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} [y_i = \hat{y}_i]$$

где N_{test} — число примеров в тестовой выборке, y_i — истинная метка, $\hat{y}_i = \arg \max_k p_{ik}$ — предсказанная метка.

Классификация по рубрикатору ГРНТИ . Государственный рубрикатор научно-технической информации (ГРНТИ) является стандартной иерархической системой классификации, используемой в России для систематизации потока научной информации. Данная задача позволяет оценить, насколько хорошо векторное представление улавливает тематическую направленность научной работы в соответствии с принятой в российской научной среде таксономией.

Рубрикатор ГРНТИ имеет три уровня иерархии. Однако анализ данных из eLibrary.ru показал, что второй и третий уровни заполнены крайне редко (для 15.37% и 0.11% статей соответственно), что делает их использование для обучения и оценки моделей ненадежным. По этой причине в задаче используется только первый, самый общий уровень классификатора. Всего на первом уровне

рубрикатора ГРНТИ представлено 90 классов, однако для формирования задачи были отобраны только те, доля которых в корпусе превышает 0.5%. Это привело к выбору 29 классов, исходное распределение которых представлено в таблице 6.

Классификация по типу публикации . Данная задача проверяет способность модели различать не тематическое содержание, а структурно-жанровые особенности научного документа. В качестве классов выступают типы публикаций, принятые в eLibrary.ru, такие как «научная статья», «материалы конференции», «обзорная статья», «краткое сообщение» и др. Умение различать эти типы важно для систем, которые могут по-разному обрабатывать, например, оригинальное исследование и обзор литературы. Исходное распределение типов публикаций в корпусе, представленное в таблице 7, является крайне несбалансированным. Для обеспечения робастности оценки, для итоговой задачи были отобраны четыре наиболее крупных и семантически различимых класса: «Научная статья», «Материалы конференции», «Обзорная статья» и «Краткое сообщение». Класс «Разное», не несущий смысловой нагрузки, был исключен.

Итоговые размеры сбалансированных выборок для каждой задачи классификации представлены в таблице 8.

Сводные результаты задач классификации приведены в таблицах 9 (русский язык) и 10 (английский язык). Сопоставление двух подзадач — принадлежности к теме (ГРНТИ) и жанрово-структурной классификации (тип публикации) — показывает различную чувствительность моделей к тематическим и стилевым признакам текста. Для подавляющего большинства моделей точность на ГРНТИ заметно выше, чем на типах публикаций; это указывает на то, что векторные представления лучше кодируют тематическое содержание, чем стиль и структуру документа. Доменно адаптированные модели на основе сигнала цитирования в среднем демонстрируют преимущество над своими версиями без дообучения: прирост для **SciRus-tiny-cite** относительно **SciRus-tiny** составляет около 4.3 п.п. на русском (50.94 против 46.63) и 3.35 п.п. на английском (49.64 против 46.29), а для **SciRus-small-cite** относительно **SciRus-small** — 2.92 п.п. на русском (51.36 против 48.44) и 1.78 п.п. на английском (49.73 против 47.95). Наименьший разрыв между русской и английской версиями наблюдается у моделей, ориентированных на многоязычное применение, тогда как выраженная асимметрия качества (резкое падение на русском при сопоставимых значениях на английском) свидетельствует

Таблица 6 — Процентная доля тем на первом уровне в рубрикаторе ГРНТИ.

| ГРНТИ | Процент |
|--|----------------|
| Полиграфия. Репрография. Фотооборудование | 13.84 |
| Медицина и здравоохранение | 10.61 |
| Образование. Педагогика | 8.46 |
| Государство и право. Юридические науки | 7.68 |
| Механика | 5.19 |
| Лингвистика | 4.31 |
| Сельское и лесное хозяйство | 3.89 |
| Биология | 2.85 |
| Клиническая медицина | 2.8 |
| Информатика | 2.25 |
| Машиностроение | 2.12 |
| Психология | 2.07 |
| Физика | 1.73 |
| Литературоведение | 1.68 |
| Строительство. Архитектура | 1.6 |
| Математика | 1.55 |
| Химия | 1.52 |
| Политика и политические науки | 1.47 |
| Автоматизация. Вычислительная техника | 1.37 |
| Физическое воспитание и спорт | 1.31 |
| Геология | 1.11 |
| Высшее профессиональное образование. Педагогика высшей школы | 0.75 |
| Искусствоведение | 0.72 |
| Транспорт | 0.7 |
| Горное дело | 0.69 |
| Энергетика | 0.68 |
| Растениеводство | 0.62 |
| Пищевая промышленность | 0.61 |
| Культурология | 0.6 |

о англоцентричном характере предобучения и недостаточном охвате русскоязычной научной лексики и стиля.

Таблица 7 — Процентная доля типов публикаций

| Тип публикации | Процент |
|-----------------------|---------|
| Научная статья | 93.98 |
| Материалы конференции | 2.15 |
| Обзорная статья | 1.98 |
| Разное | 0.67 |
| Краткое сообщение | 0.62 |
| Рецензия | 0.22 |
| Персоналия | 0.19 |

Таблица 8 — Размеры выборок для задач классификации

| Название задачи | Язык | Выборка | Количество строк |
|-----------------|------------|-----------|------------------|
| ГРНТИ | русский | обучающая | 28399 |
| | | тестовая | 2764 |
| | английский | обучающая | 24338 |
| | | тестовая | 2517 |
| Тип публикации | русский | обучающая | 4150 |
| | | тестовая | 462 |
| | английский | обучающая | 4150 |
| | | тестовая | 462 |

Таблица 9 — Результаты для задач классификации на русском языке

| Модель | ГРНТИ | Тип публикации | Среднее |
|--------------------------|--------|----------------|---------|
| gte-Qwen2-7B-instruct | 0.6767 | 0.3823 | 0.5295 |
| GritLM-7B | 0.6521 | 0.4063 | 0.5292 |
| Giga-Embeddings-instruct | 0.6638 | 0.3831 | 0.5235 |
| Linq-Embed-Mistral | 0.6406 | 0.397 | 0.5188 |
| SciRus-small-cite | 0.6641 | 0.363 | 0.5136 |
| FRIDA | 0.6611 | 0.3639 | 0.5125 |
| SFR-Embedding-Mistral | 0.6625 | 0.3621 | 0.5123 |
| SFR-Embedding-2_R | 0.6611 | 0.3604 | 0.5107 |
| SciRus-tiny-cite | 0.655 | 0.3639 | 0.5094 |

Продолжение таблицы 9

| Модель | ГРНТИ | Тип публикации | Среднее |
|---|--------|----------------|---------|
| gte-Qwen2-1.5B-instruct | 0.6511 | 0.3565 | 0.5038 |
| BERTA | 0.6486 | 0.3539 | 0.5012 |
| NV-Embed-v2 | 0.589 | 0.4017 | 0.4953 |
| multilingual-e5-large-instruct | 0.622 | 0.368 | 0.495 |
| SciRus-small | 0.6037 | 0.3652 | 0.4844 |
| jina-embeddings-v3 | 0.598 | 0.3604 | 0.4792 |
| rubert-mini-frida | 0.6066 | 0.3411 | 0.4739 |
| SciRus-tiny | 0.5804 | 0.3522 | 0.4663 |
| USER-bge-m3 | 0.5766 | 0.3489 | 0.4627 |
| USER-base | 0.5594 | 0.3567 | 0.458 |
| multilingual-e5-large | 0.5544 | 0.3615 | 0.458 |
| multilingual-e5-base | 0.5413 | 0.3645 | 0.4529 |
| rubert-tiny-turbo | 0.533 | 0.3615 | 0.4472 |
| multilingual-e5-small | 0.5318 | 0.3615 | 0.4466 |
| LaBSE-en-ru | 0.528 | 0.3634 | 0.4457 |
| paraphrase-multilingual-mpnet-base-v2 | 0.5549 | 0.3281 | 0.4415 |
| rubert-tiny2 | 0.4636 | 0.3565 | 0.41 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 0.4441 | 0.3108 | 0.3775 |
| rubert-tiny | 0.36 | 0.3221 | 0.341 |
| GIST-large-Embedding-v0 | 0.2256 | 0.2814 | 0.2535 |

Таблица 10 — Результаты для задач классификации на английском языке

| Модель | ГРНТИ | Тип публикации | Среднее |
|-------------------------|--------|----------------|---------|
| GritLM-7B | 0.6558 | 0.4297 | 0.5427 |
| gte-Qwen2-7B-instruct | 0.6888 | 0.3894 | 0.5391 |
| NV-Embed-v2 | 0.6526 | 0.4119 | 0.5323 |
| Linq-Embed-Mistral | 0.6471 | 0.4004 | 0.5238 |
| SFR-Embedding-Mistral | 0.6535 | 0.381 | 0.5172 |
| gte-Qwen2-1.5B-instruct | 0.6535 | 0.3712 | 0.5124 |

Продолжение таблицы 10

| Модель | ГРНТИ | Тип публикации | Среднее |
|---|--------|----------------|---------|
| SFR-Embedding-2_R | 0.6386 | 0.3654 | 0.502 |
| SciRus-small-cite | 0.6334 | 0.3613 | 0.4973 |
| SciRus-tiny-cite | 0.64 | 0.3528 | 0.4964 |
| GIST-large-Embedding-v0 | 0.6302 | 0.3485 | 0.4894 |
| multilingual-e5-large-instruct | 0.5999 | 0.3686 | 0.4842 |
| SciRus-small | 0.5946 | 0.3645 | 0.4795 |
| FRIDA | 0.6117 | 0.3461 | 0.4789 |
| BERTA | 0.5957 | 0.3489 | 0.4723 |
| jina-embeddings-v3 | 0.5961 | 0.3444 | 0.4703 |
| SciRus-tiny | 0.5808 | 0.345 | 0.4629 |
| Giga-Embeddings-instruct | 0.546 | 0.3723 | 0.4592 |
| USER-bge-m3 | 0.5596 | 0.3558 | 0.4577 |
| multilingual-e5-base | 0.5381 | 0.3604 | 0.4492 |
| multilingual-e5-large | 0.5428 | 0.3539 | 0.4483 |
| rubert-mini-frida | 0.5645 | 0.3297 | 0.4471 |
| paraphrase-multilingual-mpnet-base-v2 | 0.5493 | 0.329 | 0.4392 |
| multilingual-e5-small | 0.5296 | 0.3457 | 0.4377 |
| LaBSE-en-ru | 0.5124 | 0.355 | 0.4337 |
| USER-base | 0.4407 | 0.342 | 0.3913 |
| rubert-tiny-turbo | 0.4137 | 0.3671 | 0.3904 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 0.4461 | 0.3214 | 0.3838 |
| rubert-tiny | 0.4083 | 0.3398 | 0.374 |
| rubert-tiny2 | 0.3886 | 0.3433 | 0.3659 |

3.2.2 Задачи регрессии

Задачи регрессии в RuSciBench направлены на оценку способности векторных представлений кодировать информацию, позволяющую предсказывать количественные характеристики научных текстов. Оценка проводится аналогично задачам классификации: модель векторизации $f(x, \alpha)$ используется в предобученном виде для генерации векторных представлений \mathbf{v}_i для каждого документа x_i . Затем на этих векторах обучается легковесная регрессионная модель $g(\mathbf{v}, \theta)$. В качестве модели $g(\mathbf{v}_i, \theta)$ используется линейная регрессия:

$$g(\mathbf{v}_i, \theta) = \mathbf{v}_i^T \mathbf{w} + b,$$

где $\theta = \{\mathbf{w}, b\}$ — обучаемые параметры: вектор весов \mathbf{w} и свободный член b .

Процесс ее обучения заключается в минимизации среднеквадратичной ошибки по параметрам θ :

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N (y_i - g(\mathbf{v}_i, \theta))^2 \longrightarrow \min_{\theta}, \quad (3.2)$$

где y_i — истинное количественное значение для документа x_i , а $g(\mathbf{v}_i, \theta)$ — предсказанное моделью значение. В экспериментах используется реализация из библиотеки ‘scikit-learn’ (`sklearn.linear_model.LinearRegression` [68]) со стандартными параметрами.

Исходный набор данных для каждой задачи регрессии разделяется на обучающую и тестовую выборки в пропорции 90% к 10% соответственно. Поскольку целевая переменная является количественной, для обеспечения схожести ее распределений в обучающей и тестовой выборках была применена процедура стратификации по бинам: диапазон значений целевой переменной был разбит на дискретные интервалы, и разделение производилось со стратификацией по этим созданным категориям.

Для оценки качества решения регрессионной задачи используется коэффициент ранговой корреляции Кендалла в варианте τ_b . Эта мера сравнивает порядок объектов по истинным и предсказанным значениям, а не линейную зависимость между самими значениями, как корреляция Пирсона. Поэтому она лучше подходит для зашумленных показателей, таких как число цитирований, где важен прежде всего правильный порядок документов. Дополнительное преимущество

этой меры состоит в том, что после отсечения отрицательных значений она имеет единую интерпретацию: большее значение соответствует более высокому качеству. Это согласуется с мерами, используемыми в задачах классификации (Accuracy) и ранжирования (nDCG), и позволяет усреднять результаты по разнородным задачам в рамках общей оценки модели.

Коэффициент вычисляется по формуле:

$$\tau_b = \frac{P - Q}{\sqrt{(P + Q + T_{\hat{y}})(P + Q + T_y)}}, \quad (3.3)$$

где

$P = \sum_{i < j} [(y_i - y_j)(\hat{y}_i - \hat{y}_j) > 0]$ — число согласованных пар, для которых ранги истинных и предсказанных значений совпадают.

$Q = \sum_{i < j} [(y_i - y_j)(\hat{y}_i - \hat{y}_j) < 0]$ — число несогласованных пар.

$T_y = \sum_{i < j} [y_i = y_j, \hat{y}_i \neq \hat{y}_j]$ — число пар с совпадающими истинными значениями, но разными предсказанными.

$T_{\hat{y}} = \sum_{i < j} [\hat{y}_i = \hat{y}_j, y_i \neq y_j]$ — число пар с совпадающими предсказанными значениями, но разными истинными.

Коэффициент Кендалла принимает значения от -1 до 1. Отрицательное значение означает обратную ранговую зависимость между истинными и предсказанными значениями и для данной задачи не рассматривается как полезный результат. Поэтому при вычислении итоговой меры качества отрицательные значения заменяются нулем.

$$\tau_{final} = \max(0, \tau_b)$$

Предсказание числа цитирований Задача заключается в прогнозировании количества цитирований статьи другими научными работами. Это позволяет оценить, насколько хорошо векторное представление документа отражает его научную значимость и влияние. Следует отметить, что количество цитирований зависит от множества внешних факторов: не только от содержания аннотации и текста публикации, но и от авторитета автора, престижа журнала, текущих тенденций в научной области и случайных факторов. Поэтому не следует ожидать, что модели, основанные исключительно на текстовых данных, достигнут высоких значений меры качества в этой задаче. Тем не менее, эксперименты, представленные далее в работе, показывают, что качество предсказаний

значительно превосходит случайный уровень, что указывает на наличие в тексте полезного сигнала. Практика включения подобной задачи в оценочные наборы подтверждается наличием аналогичной задачи в авторитетном англоязычном бенчмарке SciDocs [22].

Размеры обучающих и тестовых выборок для задачи регрессии представлены в таблице 11.

Таблица 11 — Размеры выборок для задачи предсказания числа цитирований

| Язык | Выборка | Количество строк |
|------------|-----------|------------------|
| русский | обучающая | 164037 |
| | тестовая | 18227 |
| английский | обучающая | 164037 |
| | тестовая | 18227 |

Таблица 12 — Результаты для задач регрессии

| Модель | Количество цитат (ру.) | Количество цитат (англ.) |
|--------------------------------|------------------------|--------------------------|
| SciRus-small-cite | 0.0838 | 0.0908 |
| SciRus-small | 0.0584 | 0.0756 |
| rubert-mini-frida | 0.0691 | 0.0625 |
| multilingual-e5-small | 0.0651 | 0.0605 |
| rubert-tiny-turbo | 0.0657 | 0.0535 |
| multilingual-e5-large-instruct | 0.0596 | 0.054 |
| SciRus-tiny-cite | 0.0592 | 0.0499 |
| BERTA | 0.0544 | 0.0516 |
| multilingual-e5-large | 0.0513 | 0.0546 |
| multilingual-e5-base | 0.043 | 0.053 |
| rubert-tiny | 0.0421 | 0.0521 |
| jina-embeddings-v3 | 0.052 | 0.0421 |
| NV-Embed-v2 | 0.0438 | 0.048 |
| Linq-Embed-Mistral | 0.0364 | 0.0486 |
| USER-base | 0.0463 | 0.0384 |
| rubert-tiny2 | 0.0379 | 0.045 |

Продолжение таблицы 12

| Модель | Количество цитат (ру.) | Количество цитат (англ.) |
|---|------------------------|--------------------------|
| SciRus-tiny | 0.0312 | 0.0515 |
| gte-Qwen2-7B-instruct | 0.0405 | 0.0419 |
| paraphrase-multilingual-mpnet-base-v2 | 0.0412 | 0.0401 |
| USER-bge-m3 | 0.0385 | 0.0419 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 0.0336 | 0.0444 |
| LaBSE-en-ru | 0.0349 | 0.039 |
| SFR-Embedding-Mistral | 0.0347 | 0.034 |
| SFR-Embedding-2_R | 0.0318 | 0.0339 |
| GritLM-7B | 0.0148 | 0.0463 |
| gte-Qwen2-1.5B-instruct | 0.0157 | 0.0331 |
| FRIDA | 0.0249 | 0.0226 |
| GIST-large-Embedding-v0 | 0.011 | 0.0255 |
| Giga-Embeddings-instruct | 0.009 | 0.0 |

Итоги задач регрессии суммированы в таблице 12, где для каждой модели представлены значения меры качества для русской и английской версий задачи. Лидирующее качество показывает **SciRus-small-cite**, превосходя собственную версию без сигнала цитирования: на русском прирост составляет 2.54 п.п. (0.0838 против 0.0584), на английском — 1.52 п.п. (0.0908 против 0.0756). Отмечается языковая асимметрия: для ряда крупных универсальных моделей значения на английском стабильно выше, чем на русском, тогда как доменно адаптированные русско-английские модели удерживают паритет или превосходят на русском, что указывает на важность доменной адаптации для улавливания слабого сигнала, связанного с числом цитирований. Для **SciRus-tiny** добавление сигнала цитирования дает значительный выигрыш на русском (рост с 0.0312 до 0.0592) при почти нейтральном эффекте на английском (0.0515 против 0.0499).

3.2.3 Задачи информационного поиска

Задачи информационного поиска в рамках RuSciBench предназначены для оценки способности моделей эффективно извлекать релевантные научные документы из большого корпуса на основе текстового запроса. Постановка задачи следует стандартной парадигме информационного поиска и включает три ключевых компонента: коллекцию запросов $\mathcal{Q} = \{q_j\}_{j=1}^M$, обширный корпус документов $\mathcal{D} = \{d_i\}_{i=1}^N$, среди которых ведется поиск, и эталонные данные о релевантности, которые определяют, какие документы из корпуса являются релевантными для каждого запроса. Структура этих компонентов соответствует общепринятым форматам в таких авторитетных бенчмарках, как BEIR [26] и MTEB [25].

Процедура оценки начинается с этапа векторизации, на котором с использованием оцениваемой модели $f(x, \alpha)$ генерируются векторные представления для всех запросов из \mathcal{Q} и всех документов из \mathcal{D} . Для каждого вектора запроса \mathbf{v}_q вычисляется мера сходства с каждым вектором документа \mathbf{v}_d в корпусе. В качестве основной меры используется косинусная близость:

$$\text{similarity}(\mathbf{v}_q, \mathbf{v}_d) = \frac{\mathbf{v}_q \cdot \mathbf{v}_d}{\|\mathbf{v}_q\| \cdot \|\mathbf{v}_d\|} = \frac{\sum_{k=1}^n v_{q,k} v_{d,k}}{\sqrt{\sum_{k=1}^n v_{q,k}^2} \sqrt{\sum_{k=1}^n v_{d,k}^2}},$$

где n — размерность векторного пространства, а $v_{q,k}$ и $v_{d,k}$ — компоненты векторов запроса и документа соответственно. Некоторые модели могут быть оптимизированы для использования скалярного произведения ($\mathbf{v}_q \cdot \mathbf{v}_d$), и в таких случаях применяется мера, рекомендованная разработчиками модели. На основе вычисленных мер сходства документы корпуса ранжируются для каждого запроса.

В дополнение к моделям плотных векторных представлений был оценен лексический метод BM25 в реализации bm25s. В отличие от нейронных кодировщиков, BM25 не строит семантическое векторное пространство, а ранжирует документы по совпадениям терминов с учетом их частоты в документе и обратной документной частоты. Поэтому его результаты важны как сильная лексическая точка отсчета: они показывают, насколько задача может решаться за счет прямого терминологического совпадения между запросом и релевантным документом.

Для оценки качества ранжирования используется нормализованный дисконтированный совокупный выигрыш (Normalized Discounted Cumulative Gain,

NDCG) на первых 10 позициях, обозначаемый как $NDCG@10$. $NDCG@k$ определяется как отношение дисконтированного совокупного выигрыша (DCG) к его идеальному значению (IDCG):

$$NDCG@k = \frac{DCG@k}{IDCG@k}. \quad (3.4)$$

Дисконтированный совокупный выигрыш $DCG@k$ рассчитывается как сумма релевантностей документов, взвешенных с учетом их позиций:

$$DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i + 1)},$$

где rel_i — это оценка релевантности документа на i -й позиции. В задаче поиска цитирований релевантность бинарная: $rel_i = 1$, если документ на позиции i является релевантным (процитированным), и $rel_i = 0$ в противном случае. С учетом этого, формула $DCG@k$ упрощается до:

$$DCG@k = \sum_{i=1}^k \frac{[rel_i = 1]}{\log_2(i + 1)}.$$

Идеальный дисконтированный совокупный выигрыш $IDCG@k$ представляет собой максимально возможное значение $DCG@k$, которое достигается при идеальном ранжировании, когда все релевантные документы находятся в начале списка. Он рассчитывается как:

$$IDCG@k = \sum_{i=1}^{\min(k, |\mathcal{R}_q|)} \frac{1}{\log_2(i + 1)},$$

где $|\mathcal{R}_q|$ — общее число релевантных документов для запроса q .

Предсказание прямых цитирований В рамках RuSciBench представлена задача предсказания прямых цитирований. В терминах информационного поиска, для заданной статьи-запроса, представленной конкатенацией ее заголовка и аннотации, необходимо найти среди всех статей корпуса те, которые она цитирует. Релевантность в данном случае является бинарной: документ считается релевантным, если он процитирован статьей-запросом, и нерелевантным в противном случае.

Статистика по наборам данных для задачи поиска цитирований представлена в таблице 13.

Таблица 13 — Размеры выборок для задачи поиска цитирований

| Язык | Тип данных | Количество |
|------------|------------------|------------|
| русский | Запросы | 3000 |
| | Документы | 90000 |
| | Релевантные пары | 15000 |
| английский | Запросы | 3000 |
| | Документы | 90000 |
| | Релевантные пары | 15000 |

Таблица 14 — Результаты для задач поиска

| Модель | Цитирование (ру.) | Цитирование (англ.) |
|--------------------------------|-------------------|---------------------|
| SFR-Embedding-2_R | 0.4065 | 0.4119 |
| GritLM-7B | 0.3987 | 0.4119 |
| SFR-Embedding-Mistral | 0.3846 | 0.3782 |
| Linq-Embed-Mistral | 0.3764 | 0.3814 |
| gte-Qwen2-1.5B-instruct | 0.372 | 0.3713 |
| gte-Qwen2-7B-instruct | 0.379 | 0.3554 |
| jina-embeddings-v3 | 0.3694 | 0.3582 |
| SciRus-small-cite | 0.3598 | 0.3351 |
| SciRus-tiny-cite | 0.3508 | 0.3353 |
| Giga-Embeddings-instruct | 0.3693 | 0.3147 |
| multilingual-e5-large-instruct | 0.3467 | 0.3366 |
| multilingual-e5-large | 0.3459 | 0.3348 |
| USER-bge-m3 | 0.3517 | 0.3228 |
| BERTA | 0.3465 | 0.3165 |
| SciRus-small | 0.3328 | 0.328 |
| NV-Embed-v2 | 0.307 | 0.3433 |
| SciRus-tiny | 0.3231 | 0.3205 |
| rubert-mini-frida | 0.3201 | 0.2818 |
| multilingual-e5-base | 0.3039 | 0.2977 |
| FRIDA | 0.3016 | 0.2752 |
| multilingual-e5-small | 0.2892 | 0.2584 |
| BM25 (bm25s) | 0.2726 | 0.2738 |

Продолжение таблицы 14

| Модель | Цитирование (ру.) | Цитирование (англ.) |
|---|-------------------|---------------------|
| GIST-large-Embedding-v0 | 0.103 | 0.3894 |
| USER-base | 0.2861 | 0.1589 |
| paraphrase-multilingual-mpnet-base-v2 | 0.186 | 0.2507 |
| rubert-tiny-turbo | 0.2301 | 0.1318 |
| LaBSE-en-ru | 0.1948 | 0.166 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 0.1183 | 0.1358 |
| rubert-tiny2 | 0.1218 | 0.1033 |
| rubert-tiny | 0.0601 | 0.0732 |

Сопоставление результатов задачи поиска цитирований представлено в таблице 14. Доменно адаптированные модели с использованием данных о цитировании демонстрируют устойчивый выигрыш относительно версий без такого дообучения: для **SciRus-small** прирост на русском составляет около 2.7 п.п. (0.3598 против 0.3328), для **SciRus-tiny** — порядка 2.8 п.п. (0.3508 против 0.3231), при этом на английском выигрыш менее выражен.

Результаты BM25 показывают, что одного лексического совпадения недостаточно для достижения качества доменно адаптированных векторных моделей: метод получает $nDCG@10$, равный 0.2726 на русском и 0.2738 на английском. Это ниже не только моделей, дообученных на цитатном сигнале, но и базовых **SciRus-tiny** и **SciRus-small**. При этом близость значений для русского и английского языков показывает, что для BM25 в данной задаче различия между языками выражены слабее, чем ограничение самого лексического подхода.

3.2.4 Задачи кросс-языкового поиска

Задачи кросс-языкового поиска предназначены для оценки способности моделей сопоставлять семантически эквивалентные тексты на разных языках. Данный тип задач, известный как поиск параллельных текстов (Bitext Mining),

является стандартным подходом для оценки качества двуязычных векторных представлений. Его цель — оценить, насколько хорошо модель способна формировать общее семантическое пространство, в котором векторы текстов-переводов близки друг к другу.

В рамках RuSciBench представлена задача поиска перевода аннотации с русского языка на английский. Для ее решения используются два параллельных корпуса: исходный корпус \mathcal{D}_{ru} , содержащий 10 000 аннотаций на русском языке, и целевой корпус \mathcal{D}_{en} , содержащий их точные переводы на английский язык. Для каждой аннотации-запроса $q_j \in \mathcal{D}_{ru}$ существует ровно один верный перевод $d_{j,true} \in \mathcal{D}_{en}$.

Процедура оценки включает следующие шаги. Сначала с помощью оцениваемой модели $f(x, \alpha)$ генерируются векторные представления для всех текстов в исходном корпусе $\{\mathbf{v}_{q,j}\}$ и в целевом корпусе $\{\mathbf{v}_{d,i}\}$. Затем для каждого вектора запроса $\mathbf{v}_{q,j}$ в целевом корпусе находится вектор документа с наибольшим значением косинусной близости. Предсказанным переводом \hat{d}_j считается документ, соответствующий этому вектору (ближайший сосед):

$$\hat{d}_j = \arg \max_{d_i \in \mathcal{D}_{en}} \text{similarity}(\mathbf{v}_{q,j}, \mathbf{v}_{d,i}).$$

Качество модели оценивается путем сравнения предсказанного перевода \hat{d}_j с эталонным $d_{j,true}$ для каждого запроса из \mathcal{D}_{ru} .

В качестве меры качества используется точность (Accuracy), которая вычисляется как доля правильно идентифицированных переводов. Формально, она определяется следующим образом:

$$\text{Accuracy} = \frac{1}{|\mathcal{D}_{ru}|} \sum_{q_j \in \mathcal{D}_{ru}} [\hat{d}_j = d_{j,true}]$$

где $|\mathcal{D}_{ru}|$ — общее число запросов, \hat{d}_j — предсказанный перевод для запроса q_j , $d_{j,true}$ — истинный перевод. Эта задача является важным тестом способности модели улавливать семантическую эквивалентность текстов вне зависимости от языка, что критично для построения эффективных многоязычных систем в научной сфере.

Таблица 15 — Результаты для задачи поиска перевода

| Модель | русский-английский |
|---|--------------------|
| SFR-Embedding-2_R | 0.9992 |
| GritLM-7B | 0.9989 |
| SFR-Embedding-Mistral | 0.9987 |
| gte-Qwen2-1.5B-instruct | 0.9984 |
| Linq-Embed-Mistral | 0.9981 |
| jina-embeddings-v3 | 0.9979 |
| multilingual-e5-large-instruct | 0.9979 |
| multilingual-e5-large | 0.9971 |
| USER-bge-m3 | 0.9964 |
| BERTA | 0.9951 |
| Giga-Embeddings-instruct | 0.9947 |
| gte-Qwen2-7B-instruct | 0.9932 |
| multilingual-e5-base | 0.9914 |
| NV-Embed-v2 | 0.9907 |
| LaBSE-en-ru | 0.9887 |
| FRIDA | 0.9873 |
| SciRus-small | 0.9866 |
| SciRus-tiny | 0.9852 |
| rubert-mini-frida | 0.9805 |
| multilingual-e5-small | 0.9748 |
| SciRus-small-cite | 0.96 |
| SciRus-tiny-cite | 0.953 |
| paraphrase-multilingual-mpnet-base-v2 | 0.9321 |
| rubert-tiny-turbo | 0.8644 |
| USER-base | 0.8509 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 0.7041 |
| rubert-tiny2 | 0.675 |
| rubert-tiny | 0.436 |
| GIST-large-Embedding-v0 | 0.0456 |

Результаты для задачи кросс-языкового поиска перевода приведены в таблице 15. Верхняя группа моделей демонстрирует насыщение меры качества

вплотную к 100%, что указывает на низкую сложность идентификации точных переводов аннотаций при использовании современных кодировщиков и на то, что задача ближе к проверке корректности выравнивания базового семантического пространства, чем к тонкой дифференциации качества представлений. Различия между моделями общего назначения и доменно адаптированными кодировщиками здесь существенно меньше, чем в задачах поиска и регрессии, что дополнительно подтверждает близость задачи насыщению качества на рассматриваемом корпусе.

3.3 Оценка моделей на RuSciBench

Итоговая таблица 16 представляет собой сводный рейтинг моделей, оцененных на бенчмарке RuSciBench. Для агрегирования результатов используется метод Борда, заимствованный из теории социального выбора и хорошо зарекомендовавший себя как робастный способ сравнения систем обработки естественного языка [69]. Суть метода заключается в том, что каждая задача бенчмарка рассматривается как «голосующий», который ранжирует все модели в соответствии с их производительностью. Модель получает баллы в зависимости от занятого места: чем выше ранг, тем больше баллов. Итоговый балл модели B_i формируется как сумма баллов, полученных по всем задачам. Формально, он вычисляется следующим образом:

$$B_i = \sum_{j=1}^m (n - r_{ij}), \quad (3.5)$$

где m — общее число задач, n — количество сравниваемых моделей, а r_{ij} — ранг i -й модели в j -й задаче (где ранг 1 является наилучшим). Итоговый рейтинг моделей строится на основе убывания их суммарных баллов B_i . Важно отметить, что итоговый ранг по методу Борда, а также представленные в таблице средние значения мер качества, вычислены по всему набору из 18 задач бенчмарка.

Таблица 16 — Сводный рейтинг моделей на RuSciBench

| Модель | Ранг по методу Борда | Среднее по русскоязычным задачам | Среднее по англоязычным задачам |
|--------------------------------|----------------------|----------------------------------|---------------------------------|
| GritLM-7B | 1 | 0.3873 | 0.4176 |
| Linq-Embed-Mistral | 2 | 0.3795 | 0.4001 |
| SFR-Embedding-Mistral | 3 | 0.3811 | 0.3946 |
| SFR-Embedding-2_R | 4 | 0.3878 | 0.3983 |
| gte-Qwen2-7B-instruct | 5 | 0.3851 | 0.4 |
| SciRus-small-cite | 6 | 0.3938 | 0.3903 |
| multilingual-e5-large-instruct | 7 | 0.3693 | 0.3702 |
| SciRus-tiny-cite | 8 | 0.3861 | 0.3837 |
| NV-Embed-v2 | 9 | 0.3554 | 0.3978 |
| gte-Qwen2-1.5B-instruct | 10 | 0.3655 | 0.3868 |
| BERTA | 11 | 0.3771 | 0.3641 |
| SciRus-small | 12 | 0.3583 | 0.3699 |
| jina-embeddings-v3 | 13 | 0.3621 | 0.3609 |
| Giga-Embeddings-instruct | 14 | 0.3608 | 0.3144 |
| multilingual-e5-large | 15 | 0.3497 | 0.3515 |
| rubert-mini-frida | 16 | 0.3675 | 0.347 |
| SciRus-tiny | 17 | 0.3542 | 0.3667 |
| FRIDA | 18 | 0.3566 | 0.339 |
| multilingual-e5-base | 19 | 0.3422 | 0.3475 |
| USER-bge-m3 | 20 | 0.3502 | 0.3468 |
| multilingual-e5-small | 21 | 0.3437 | 0.3377 |
| rubert-tiny-turbo | 22 | 0.3333 | 0.2885 |
| LaBSE-en-ru | 23 | 0.3083 | 0.306 |
| USER-base | 24 | 0.3404 | 0.2858 |
| rubert-tiny2 | 25 | 0.2881 | 0.2703 |
| GIST-large-Embedding-v0 | 26 | 0.194 | 0.3762 |

Продолжение таблицы 16

| Модель | Ранг по методу Борда | Среднее по русскоязычным задачам | Среднее по англоязычным задачам |
|---|----------------------|----------------------------------|---------------------------------|
| rubert-tiny | 27 | 0.245 | 0.2745 |
| paraphrase-multilingual-mpnet-base-v2 | 28 | 0.3045 | 0.321 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 29 | 0.2543 | 0.2698 |

После представления общего рейтинга моделей, целесообразно проанализировать факторы, влияющие на их производительность. Одним из ключевых факторов является размер модели, или количество ее параметров. На рисунке 3.1 визуализирована зависимость среднего результата модели на всех задачах бенчмарка RuSciBench от ее размера.

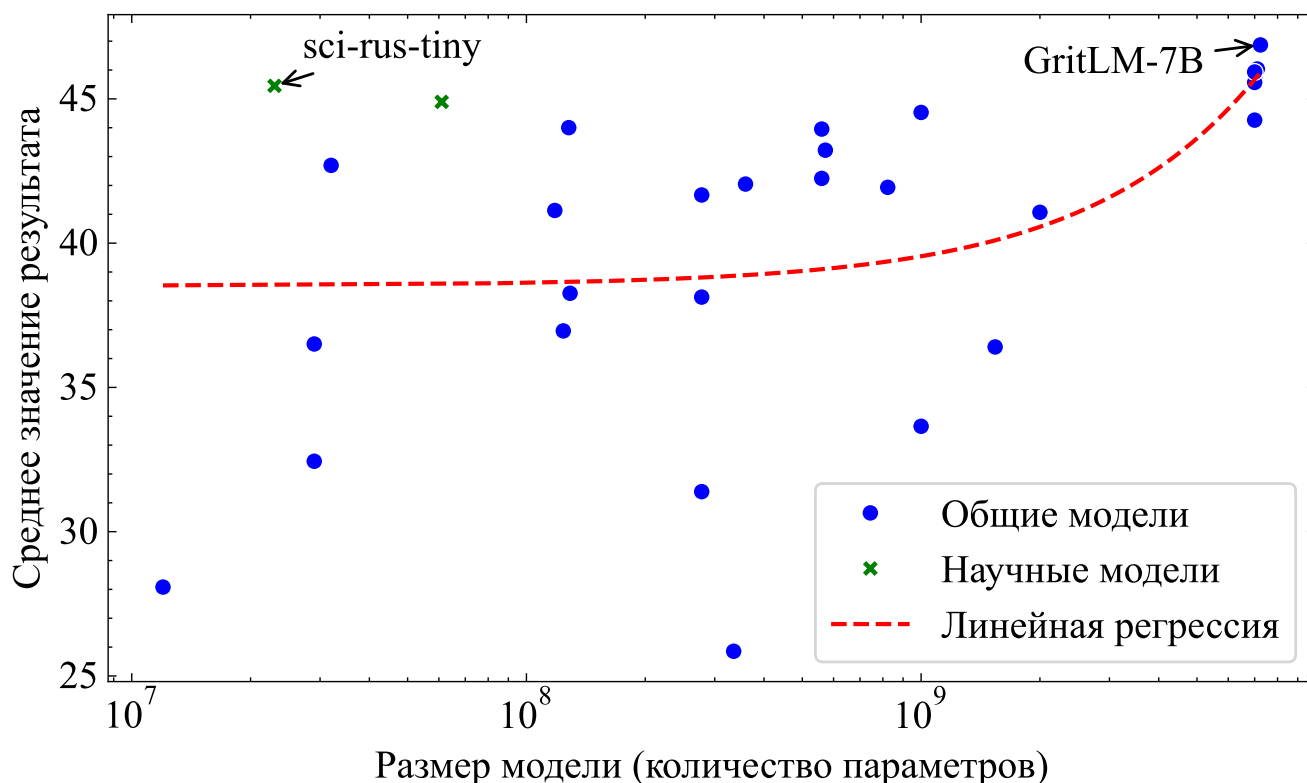


Рисунок 3.1 — Зависимость среднего результата на всех задачах от размера модели.

На рисунке 3.1 представлена зависимость усредненного по всем задачам RuSciBench показателя качества модели от ее размера, измеряемого количе-

ством параметров. Ось абсцисс, представляющая размер модели, использует логарифмическую шкалу для охвата широкого диапазона значений — от десятков миллионов до нескольких миллиардов параметров. Ось ординат отражает среднее значение меры качества по всем русскоязычным и англоязычным задачам бенчмарка.

Анализ графика позволяет выявить общую тенденцию к увеличению производительности моделей с ростом их размера. Большинство моделей общего назначения (обозначены синими точками) демонстрируют положительную корреляцию между количеством параметров и средним результатом. Эта тенденция аппроксимирована пунктирной линией регрессии, которая показывает монотонный рост среднего качества с увеличением масштаба модели. Наиболее крупные модели общего назначения, такие как GritLM-7B (7.24 млрд параметров), располагаются в верхней правой части графика, достигая одних из самых высоких средних показателей качества среди всех оцененных моделей.

Вместе с тем зависимость качества от размера модели не является однозначной. Среди моделей со сходным числом параметров, особенно в диапазоне от 10^8 до 10^9 , наблюдается заметный разброс результатов. Следовательно, на качество влияют не только масштаб модели, но и архитектура, состав обучающих данных, способ оптимизации и стратегия дообучения. Полученные результаты согласуются с представлением о пользе масштабирования, но показывают, что для обработки русскоязычных научных текстов важны также доменная адаптация и языковой состав обучающего корпуса.

3.3.1 Оценка степени языковой специализации моделей

Бенчмарк RuSciBench имеет парную структуру: каждая задача и каждый текстовый пример на русском языке имеют семантически эквивалентную англоязычную версию. В такой постановке многоязычная модель с устойчивым кросс-языковым поведением должна показывать близкое качество на обеих языковых версиях задач. Систематическое отклонение от этого баланса указывает на языковую специализацию модели, то есть на ее лучшую адаптацию к одному из языков.

Для количественной оценки языковой специализации в таблице 17 сопоставлены результаты каждой модели на русском и английском языках. Вместо одного общего ранга вычисляются два независимых рейтинга: «Ранг Борда (рус.)» и «Ранг Борда (англ.)». Первый строится по русскоязычным задачам, второй по англоязычным. Такое разделение показывает, как меняется положение модели в рейтинге при переходе от одного языка к другому.

Ключевым показателем является столбец «Прирост на рус. по отношению к англ. (%)». Он отражает процентное изменение средней меры качества модели при переходе от английских задач к русским и тем самым показывает направление и величину языковой специализации.

Интерпретация значений в этом столбце является следующей:

- **Положительное значение** означает, что модель показывает более высокое качество на русскоязычных задачах.
- **Отрицательное значение** говорит о специализации модели на английском языке, поскольку на нем достигаются более высокие результаты.
- **Значение, близкое к нулю**, соответствует более сбалансированному качеству на двух языках.
- **Абсолютная величина значения** отражает степень различия между языковыми версиями задач. Большие по модулю значения указывают на выраженный перекося качества.

В таблице модели упорядочены в соответствии с общим рейтингом из таблицы 16. Это позволяет сопоставить общий уровень качества модели с ее языковым балансом.

Таблица 17 — Оценка степени языковой специализации моделей

| Модель | Количество параметров | Ранг Борда (англ.) | Ранг Борда (рус.) | Прирост на рус. по отношению к англ. (%) |
|--------------------------|-----------------------|--------------------|-------------------|--|
| GritLM-7B | 7.24 млрд | 1 | 3 | -7.26 |
| Linq-Embed-Mistral | 7.0 млрд | 2 | 7 | -5.15 |
| SFR-Embedding-Mistral | 7.0 млрд | 5 | 5 | -3.42 |
| SFR-Embedding-2_R | 7.11 млрд | 6 | 6 | -2.65 |
| gte-Qwen2-7B-instruct | 7.0 млрд | 4 | 4 | -3.74 |
| SciRus-small-cite | 61 млн | 7 | 1 | 0.89 |

Продолжение таблицы 17

| Модель | Количество параметров | Ранг Борда (англ.) | Ранг Борда (рус.) | Прирост на рус. по отношению к англ. (%) |
|---|-----------------------|--------------------|-------------------|--|
| multilingual-e5-large-instruct | 560 млн | 11 | 8 | -0.23 |
| SciRus-tiny-cite | 23 млн | 10 | 2 | 0.61 |
| NV-Embed-v2 | 7.0 млрд | 3 | 14 | -10.65 |
| gte-Qwen2-1.5B-instruct | 1.0 млрд | 9 | 17 | -5.5 |
| BERTA | 128 млн | 12 | 9 | 3.58 |
| SciRus-small | 61 млн | 11 | 13 | -3.12 |
| jina-embeddings-v3 | 572 млн | 16 | 13 | 0.32 |
| rubert-mini-frida | 32 млн | 15 | 12 | 5.92 |
| SciRus-tiny | 23 млн | 13 | 19 | -3.42 |
| multilingual-e5-large | 560 млн | 14 | 18 | -0.51 |
| Giga-Embeddings-instruct | 2.0 млрд | 22 | 10 | 14.73 |
| multilingual-e5-base | 278 млн | 16 | 19 | -1.54 |
| FRIDA | 823 млн | 21 | 15 | 5.19 |
| USER-bge-m3 | 359 млн | 20 | 22 | 1.0 |
| multilingual-e5-small | 118 млн | 18 | 19 | 1.76 |
| rubert-tiny-turbo | 29 млн | 18 | 16 | 15.56 |
| LaBSE-en-ru | 129 млн | 24 | 24 | 0.76 |
| USER-base | 124 млн | 28 | 21 | 19.1 |
| rubert-tiny2 | 29 млн | 26 | 23 | 6.55 |
| rubert-tiny | 12 млн | 23 | 26 | -10.75 |
| paraphrase-multilingual-mpnet-base-v2 | 278 млн | 27 | 25 | -5.15 |
| GIST-large-Embedding-v0 | 335 млн | 13 | 29 | -48.43 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 278 млн | 29 | 28 | -5.77 |

Результаты показывают, что модели существенно различаются по языковому балансу. Некоторые модели, занимающие высокие места в общем рейтинге, например GritLM-7B и NV-Embed-v2, получают этот результат прежде всего за счет высокого качества на английском языке. На русскоязычных задачах их среднее качество ниже на 7.26% и 10.65% соответственно. Это указывает на заметное влияние языкового состава обучающих данных и на ограниченную переносимость качества на русскоязычный научный материал. Модели семейства SciRus, несмотря на существенно меньшее число параметров, занимают первые два места в рейтинге для русского языка и демонстрируют близкие результаты на двух языках: +0.89% для SciRus-small-cite и +0.61% для SciRus-tiny-cite. Отдельные модели показывают выраженную языковую асимметрию, например GIST-large-Embedding-v0 снижает качество на русском языке на 48.43%, тогда как multilingual-e5-large-instruct имеет почти равные результаты на двух языках (-0.23%). Следовательно, при выборе модели для русскоязычных научных текстов недостаточно учитывать только общий рейтинг. Необходимо также оценивать языковой баланс, поскольку компактная доменно адаптированная модель может быть предпочтительнее более крупной модели с выраженной языковой асимметрией.

3.4 Интеграция в международный бенчмарк МТЕВ

Для обеспечения воспроизводимости и сопоставимости результатов, полученных на бенчмарке RuSciBench, с результатами других исследователей, была проведена его интеграция в ведущий международный бенчмарк Massive Multilingual Text Embedding Benchmark (МТЕВ) [38]. МТЕВ представляет собой общепринятый стандарт и открытую программную библиотеку для оценки качества моделей векторного представления текстов. Он агрегирует большое количество разнородных задач на множестве языков, что позволяет унифицировать процедуру тестирования и проводить комплексное сравнение моделей. Включение RuSciBench в МТЕВ позволяет не только валидировать разработанный инструментарий силами международного научного сообщества, но и включить русскоязычный научный домен в глобальный контекст оценки нейросетевых моделей.

Интеграция RuSciBench в экосистему МТЕВ предоставляет ряд ключевых преимуществ. Во-первых, она существенно упрощает процесс оценки для сторонних исследователей, которым достаточно использовать стандартный программный интерфейс МТЕВ, чтобы протестировать свои модели на всём наборе задач RuSciBench. Во-вторых, это позволяет проводить прямое и объективное сопоставление как специализированных моделей, ориентированных на научный домен, так и мощных многоязычных моделей общего назначения, на единой и стандартизированной платформе. Бенчмарк RuSciBench, включающий 18 задач, составляет существенную долю от общего числа задач в ММТЕВ, насчитывавшего на момент интеграции 73 задачи, что подчёркивает значимость данного вклада в развитие инструментов оценки.

Практическая реализация оценки с использованием данной интеграции демонстрирует её простоту и доступность. Процесс запуска полного цикла тестирования произвольной модели, доступной на платформе Hugging Face, на всех задачах бенчмарка RuSciBench может быть выполнен с помощью кода, приведенного в листинге 3.1.

Листинг 3.1 Пример использования бенчмарка RuSciBench в МТЕВ

```
import mteb

model_name = "mlsa-iai-msu-lab/sci-rus-tiny"
benchmark = mteb.get_benchmark("RuSciBench")
model = mteb.get_model(model_name)
evaluation = mteb.MTEB(tasks=benchmark.tasks)
results = evaluation.run(
    model,
    output_folder=f"results/{model_name.replace('/', '__')}"
)
```

Данный фрагмент кода автоматически загружает указанную модель, получает полный набор задач из бенчмарка RuSciBench, выполняет оценку по всем предопределённым мерам качества и сохраняет результаты в структурированном виде. Такая автоматизация и стандартизация являются необходимым условием для построения воспроизводимой и прозрачной системы оценки качества моделей, что, в свою очередь, способствует ускорению научного прогресса в данной области.

3.5 Основные выводы

В результате разработки и апробации RuSciBench был создан стандартизированный инструмент для оценки моделей векторного представления научных текстов на русском и английском языках. Бенчмарк включает 18 задач, охватывающих классификацию, регрессию, информационный поиск и кросс-языковое сопоставление текстов. Использование реальных метаданных научных публикаций из eLibrary.ru позволяет оценивать модели в условиях, близких к практическим сценариям работы с научной информацией.

Оценка моделей разного масштаба, от компактных кодировщиков до многомиллиардных архитектур, показала, что увеличение числа параметров в среднем связано с ростом качества. Вместе с тем результаты показывают значимость доменной специализации: модели, дообученные на научном корпусе (sci-rus-*), демонстрируют высокое качество даже при сравнительно малом числе параметров и конкурируют с крупными моделями общего назначения.

Итоговый рейтинг, построенный с использованием робастной агрегации рангов Борда, может использоваться как ориентир при выборе модели с учетом требований к качеству и вычислительным ресурсам. RuSciBench восполняет недостаток открытых инструментов оценки для русскоязычного научного домена и предоставляет основу для дальнейших исследований по разработке и адаптации моделей векторизации научных текстов.

Глава 4. Бенчмарк для оценки качества верификации научных фактов на русском языке

В предыдущих главах модели оценивались на задачах, основанных на метаданных научных публикаций, включая рубрикацию и число цитирований. Такие данные характеризуют документ в целом, но не позволяют проверить, насколько модель способна работать с содержанием конкретного научного утверждения. Рост объема научной информации, особенно в периоды глобальных кризисов, таких как пандемии, делает актуальной задачу автоматизированной проверки научных утверждений [28]. Эта задача известна как верификация научных фактов. Она требует от модели не только тематического сопоставления текстов, но и поиска релевантных доказательств, а также определения того, подтверждает или опровергает найденный текст заданное утверждение.

Для русскоязычного научного домена инструменты, позволяющие проводить такую детальную оценку, до сих пор отсутствовали. Данная глава посвящена восполнению этого пробела путем разработки нового инструментария для оценки качества моделей — бенчмарка RuSciFact. В главе детально описывается методология его полуавтоматического формирования, основанная на генерации утверждений с помощью больших языковых моделей и их последующей экспертной валидации. На основе созданного бенчмарка проводится комплексное экспериментальное исследование, в рамках которого оценивается и сравнивается производительность современных моделей векторизации, включая разработанные в диссертации модели семейства SciRus.

4.1 Постановка задачи верификации научных фактов

Задача верификации научных фактов состоит в поиске релевантного для утверждения документа и определении на его основе того, подтверждается или опровергается данное утверждение информацией, содержащейся в документе. Формально, дано:

- Утверждение c — атомарный, проверяемый научный факт.
- Корпус документов $D = \{e_i\}_{i=1}^{N_D}$ — множество аннотаций научных статей.

- Множество меток $L = \{\text{ПОДТВЕРЖДАЕТ, ОПРОВЕРГАЕТ}\}$, отражающих отношение между утверждением и аннотацией.

Итоговая цель состоит в построении модели M , которая для заданного утверждения c и корпуса D находит релевантный документ $e^* \in D$ и определяет его отношение $l^* \in L$ к утверждению:

$$(e^*, l^*) = M(c, D).$$

Для решения данной задачи с помощью моделей семантического векторного представления текста, ее целесообразно декомпозировать на два последовательных этапа: информационный поиск и классификация.

Этап 1: Информационный поиск релевантной аннотации. На первом этапе для заданного утверждения c , выступающего в роли запроса, необходимо найти наиболее релевантную аннотацию e^* в корпусе D . Эта задача решается путем отображения всех текстов в единое векторное пространство с помощью модели-кодировщика $f(x, \alpha)$ и последующего ранжирования аннотаций по мере близости к вектору утверждения.

$$e^* = \operatorname{argmax}_{e_i \in D} s(f(c, \alpha), f(e_i, \alpha)), \quad (4.1)$$

где $f(x, \alpha)$ — параметризованная модель семантического векторного представления текста с параметрами α , а $s(\cdot, \cdot)$ — функция близости векторов, как правило, косинусная.

Качество решения этой задачи оценивается с помощью меры Mean Reciprocal Rank (MRR@k), которая вычисляется как среднее обратных рангов первой релевантной аннотации:

$$\text{MRR@k} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{[\text{rank}_i \leq k]}{\text{rank}_i},$$

где Q — множество запросов (утверждений), а rank_i — ранг первого правильного документа для i -го запроса (если ранг $> k$, то слагаемое считается равным нулю).

Этап 2: Классификация пары «утверждение–аннотация». На втором этапе для найденной или предоставленной пары (c, e) решается задача бинарной классификации. Для этой подзадачи в работе рассматриваются два подхода. Основной

подход, основанный на векторных моделях, предполагает обучение легковесного классификатора $g(\cdot, \theta)$ поверх замороженных представлений $f(c, \alpha)$ и $f(e, \alpha)$. Процесс обучения сводится к минимизации функции потерь, в качестве которой используется бинарная перекрестная энтропия:

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log p_i + (1 - y_i) \log(1 - p_i)] \rightarrow \min_{\theta}, \quad (4.2)$$

где N — число пар в обучающей выборке, $y_i \in \{0, 1\}$ — истинная метка для i -й пары (1 для ПОДТВЕРЖДАЕТ и 0 для ОПРОВЕРГАЕТ), а p_i — вероятность принадлежности к классу ПОДТВЕРЖДАЕТ, предсказанная моделью g .

Анализ существующих наборов данных показывает отсутствие инструментов для оценки моделей в рамках такой постановки для русскоязычных научных текстов. Это определяет научную задачу данной главы — разработку и апробацию такого инструментария.

4.2 Методология формирования набора данных RuSciFact

В качестве исходного материала для формирования набора данных был использован корпус аннотаций из бенчмарка RuSciBench [36], детально описанного в Главе 3. Данный выбор обусловлен тем, что RuSciBench содержит обширный и репрезентативный срез русскоязычных научных публикаций из различных дисциплин, а сами тексты уже прошли этап предварительной очистки. Это позволило сфокусироваться непосредственно на основной задаче — генерации и верификации научных фактов, опираясь на подготовленную и верифицированную текстовую базу.

Ключевым методологическим решением стало использование большой языковой модели для автоматической генерации научных утверждений. Эффективность такого подхода для ускорения и масштабирования процесса создания наборов данных продемонстрирована в ряде недавних исследований [70; 71]. Выбор конкретной генеративной модели, однако, является нетривиальной задачей, требующей объективных критериев оценки. Обоснование выбора модели было сделано на основе результатов комплексного русскоязычного бенчмарка MERA (Multitask Evaluation of Russian-language Abilities) [72]. Данный бенчмарк

представляет собой многозадачный оценочный набор, предназначенный для всестороннего анализа способностей языковых моделей в обработке русского языка. Он включает в себя широкий спектр задач, таких как ответы на вопросы, классификация текстов, семантическая близость, суммаризация и логический вывод. Задача генерации научного утверждения на основе аннотации требует от модели комплекса способностей: глубокого понимания текста, способности к логическому умозаключению и навыков синтеза нового текста. Поскольку бенчмарк MERA оценивает именно эти фундаментальные способности, высокий совокупный результат модели на нем является надежным индикатором ее пригодности для решения данной, более узкой и специфической, задачи.

Таблица 18 — Результаты оценки языковых моделей на бенчмарке MERA (агрегированная мера качества). Данные приведены по состоянию на 2 ноября 2024 года [72].

| Модель | Открытые веса | Результат |
|-------------------------------------|---------------|--------------|
| Human Benchmark | - | 0.852 |
| GPT-4o | нет | 0.642 |
| Meta-Llama-3.1-405B-Instruct | да | 0.590 |
| GigaChat Max | нет | 0.588 |
| Mistral-Large-Instruct-2407 | да | 0.574 |
| GPT-4o-mini | нет | 0.570 |
| Qwen2-72B-Instruct | да | 0.570 |
| Meta-Llama-3.1-70B-Instruct | да | 0.554 |

Результаты, представленные в Таблице 18, показывают, что на момент проведения работы модель Meta-Llama-3.1-405B-Instruct¹ продемонстрировала наилучшие результаты среди всех моделей с открытыми весами, что является ключевым фактором для обеспечения воспроизводимости данного исследования. Исходя из этого, именно данная модель была выбрана в качестве основного инструмента для генерации утверждений в рамках конвейера RuSciFact.

Для применения модели использовался вычислительный кластер, состоящий из восьми графических ускорителей NVIDIA A100 с объемом видеопамати 80 ГБ каждый. Развертывание и эффективное исполнение запросов к модели обеспечивалось с помощью специализированной библиотеки vLLM [73]. Полная

¹<https://huggingface.co/meta-llama/Llama-3.1-405B-FP8>

версия модели, использующая 16-битное представление весов, требует для размещения в памяти около 810 ГБ (405×10^9 параметров \times 2 байта/параметр), что превышает суммарный объем памяти доступного кластера (640 ГБ). В связи с этим, для проведения экспериментов была задействована версия модели, квантованная до 8 бит. Такое преобразование позволяет сократить требования к памяти вдвое, до приблизительно 405 ГБ, делая возможным ее размещение на используемом оборудовании без существенной потери качества генерации[74].

Для того чтобы генерируемые данные были релевантны поставленной задаче и позволяли проводить содержательную оценку моделей, был сформулирован ряд требований, которые легли в основу инструкций (промптов) для LLM. Каждая итоговая пара «утверждение–аннотация» (c, e) должна была удовлетворять следующему своду правил.

Сначала определялись требования к аннотации-источнику e . **Информативность** являлась необходимым условием, то есть текст должен был содержать достаточный объем фактической информации для формулирования проверяемого научного вывода. Аннотации, описывающие исключительно структуру работы или общие рассуждения без конкретных результатов, отбраковывались. Примером такой неинформативной аннотации, не содержащей конкретных выводов, может служить следующий текст:

Статья посвящена традиционному объекту социально-экономической географии – районному центру. В статье описан новый подход к оценке центральных функций районных центров Ивановской области. Основу данной методики составила информация о некоторых государственных учреждениях, размещенных в райцентре. В результате анализа системы иерархии государственных учреждений были выделены несколько типов райцентров.

Далее, для информативных аннотаций, были сформулированы жесткие требования к генерируемому утверждению c :

- **Логическая выводимость:** Утверждение должно являться строгим логическим следствием текста аннотации. Это центральное требование, гарантирующее, что отношение между c и e является именно логическим, а не ассоциативным или тематическим.
- **Обоснованность:** Утверждение не должно содержать никакой информации, отсутствующей в аннотации. Каждая его часть должна быть

полностью основана на исходном тексте, исключая любые внешние знания, интерпретации или допущения.

- **Неявность:** Утверждение не должно быть результатом прямого копирования фрагмента текста. Оно должно представлять собой результат умозаключения, обобщения или переформулирования, т.е. следовать из аннотации, но не быть в ней эксплицитно указанным в той же форме. Это требование направлено на то, чтобы модели-оценщики демонстрировали способность к пониманию, а не просто к поиску подстроки.
- **Конкретность и атомарность:** Утверждение должно быть сформулировано точно, выражать одну законченную мысль и не содержать неопределенных выражений (например, «с определенными свойствами», «в некоторых состояниях») или модальных конструкций.
- **Автономность:** Утверждение должно быть полностью самодостаточным и не содержать прямых ссылок на исходный текст (например, «в данной работе», «предложенный метод», «авторы показали»). Это обеспечивает возможность его оценки в отрыве от контекста аннотации.

Соблюдение этого свода правил при генерации и последующей фильтрации данных позволило создать набор пар (c, e) , проверка которых требует от моделей не простого лексического сопоставления, а развитой способности к семантическому анализу и логическому выводу. На основе этих формализованных требований был разработан многоэтапный конвейер генерации и фильтрации данных.

4.3 Конвейер генерации и фильтрации данных

Для формирования набора данных RuSciFact был разработан и применен многоэтапный конвейер, который схематично представлен на Рисунке 4.1. Процесс был разделен на две независимые ветви: создание подтверждающих («положительных») и противоречащих («отрицательных») утверждений. Такой подход позволил тонко настроить инструкции и критерии фильтрации для каждого типа данных, учитывая их специфику.

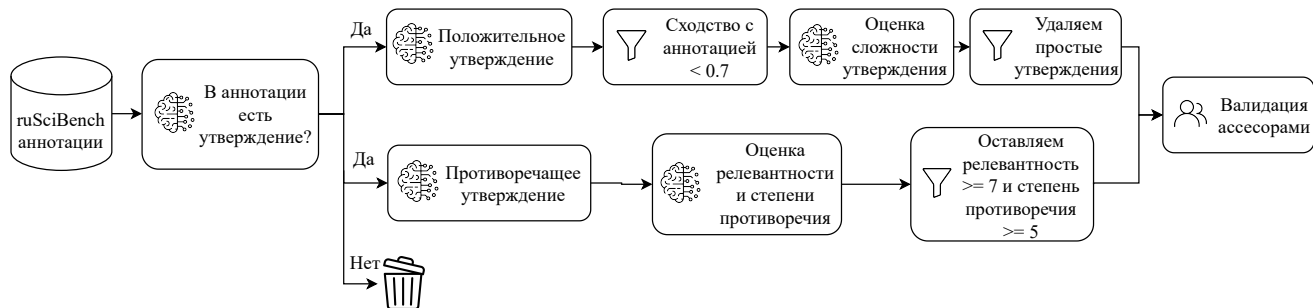


Рисунок 4.1 — Обзор конвейера генерации и валидации данных для RuSciFact.

4.3.1 Генерация подтверждающих утверждений

Создание качественных подтверждающих утверждений потребовало последовательного применения нескольких этапов фильтрации, каждый из которых был направлен на повышение валидности и сложности итогового набора данных.

Этап 1: Отбор информативных аннотаций и первичная генерация. Начальный этап был посвящен решению фундаментальной проблемы: значительная доля аннотаций в научных корпусах, включая RuSciBench, носит описательный, а не фактологический характер. Они могут описывать цели, методы или структуру исследования, не содержа при этом конкретных, проверяемых выводов. Использование таких текстов в качестве источника привело бы к генерации либо нерелевантных, либо ложных утверждений.

Для решения этой проблемы была применена стратегия предварительной фильтрации с помощью самой языковой модели. Был разработан специальный промпт, который инструктировал модель Meta-Llama-3.1-405B-Instruct выполнить одну из двух задач: либо сгенерировать научное утверждение, строго следующее из текста, либо, если это невозможно, вернуть специальный маркер «Аннотация не содержит факт». Такая постановка задачи позволила эффективно отсеять неинформативные аннотации на ранней стадии, что существенно повысило качество и релевантность первичного набора сгенерированных утверждений. По оценке модели, лишь около 42% исходных аннотаций содержали достаточный объем информации для формулирования факта. Полный текст инструкции приведён в Приложении [A.1](#).

Этап 2: Фильтрация по лексическому сходству. Следующей задачей было исключение утверждений, которые, хотя и являлись формально корректными, по сути представляли собой прямое или незначительно измененное цитирование фрагментов исходной аннотации. Наличие таких примеров в наборе данных сместило бы фокус оценки с семантического понимания на поверхностное сопоставление строк.

Чтобы гарантировать, что задача требует именно логического вывода, а не поиска подстроки, был внедрен этап фильтрации на основе лексического сходства. Для каждой пары «утверждение–аннотация» вычислялась мера сходства с использованием функции `partial_ratio`, которая оценивает степень совпадения наилучшим образом выровненных подстрок. Утверждения, для которых значение этой меры превышало эмпирически подобранный порог 0.7, отбраковывались. Этот шаг позволил целенаправленно удалить «тривиальные» примеры, оставив те, что требуют от модели анализа связей между несколькими частями текста или умозаключения.

Этап 3: Фильтрация по уровню сложности. Предварительный ручной анализ показал, что даже после предыдущих этапов фильтрации среди сгенерированных утверждений встречались общеизвестные факты, для проверки которых не требуется научный контекст. Примером может служить утверждение «Россия является самой большой страной по территории». Наличие таких примеров упрощало бы задачу и снижало бы способность бенчмарка выявлять различия между моделями при работе с научным содержанием.

Для повышения сложности и специфичности набора данных был добавлен этап классификации утверждений по уровню их нетривиальности. Языковая модель, выступая в роли эксперта, классифицировала каждое утверждение как «простое», «среднее» или «сложное» на основе промпта с четкими определениями и примерами для каждой категории. Для дальнейшей работы отбирались только утверждения средней и высокой сложности. Этот этап обеспечил фокусировку бенчмарка на проверке фактов, требующих анализа именно научного контекста, представленного в аннотации. Текст инструкции для этой классификации приведён в Приложении [A.2](#).

4.3.2 Генерация противоречащих утверждений

Генерация осмысленных и нетривиальных противоречий является значительно более сложной задачей, чем подтверждение, поскольку требует не только понимания исходного текста, но и способности формулировать альтернативные, но при этом релевантные гипотезы.

В качестве исходного материала для этого этапа использовались исключительно те аннотации, для которых на предыдущем шаге удалось успешно сгенерировать подтверждающее утверждение. Такой подход был выбран из соображений эффективности и качества. Аннотации, из которых удалось извлечь подтверждаемый вывод, по определению являются информационно насыщенными и содержат четко сформулированную основную мысль. Это делает их хорошими кандидатами для формулирования осмысленного семантического противоречия, поскольку существует конкретный тезис, который можно опровергнуть. Использование этого отфильтрованного подмножества позволило сфокусировать генерацию на заведомо качественных источниках и повысить вероятность получения валидных отрицательных примеров.

Этап 1: Формулирование семантического противоречия. Простейший способ создать противоречие — добавить частицу «не» или иное прямое отрицание. Однако такие примеры проверяют лишь способность модели распознавать синтаксическое отрицание, а не глубокое семантическое противоречие.

Чтобы создать действительно сложные «отрицательные» примеры, был разработан промпт, который явно запрещал модели использовать прямое отрицание. Вместо этого модель должна была сгенерировать утверждение, которое по смыслу является антонимичным исходному выводу аннотации. Например, если в аннотации утверждается, что «метод А эффективнее метода Б», то требуемое противоречие не должно было звучать как «метод А не эффективнее метода Б». Этот подход позволил создать примеры, для распознавания которых требуется семантический, а не синтаксический анализ. Полная формулировка инструкции приведена в Приложении [А.3](#).

Этап 2: Фильтрация с использованием модели-оценщика. При свободной генерации противоречий часть утверждений оказывается нерелевантной исход-

ной аннотации или не образует с ней прямого смыслового конфликта. Полная ручная проверка таких примеров потребовала бы значительных трудозатрат.

Для предварительного контроля качества была использована схема «LLM как судья» (*LLM-as-a-judge*)[75]. Для каждой сгенерированной пары «утверждение–аннотация» модель оценивала по 10-балльной шкале два свойства: релевантность утверждения теме аннотации и степень его поддержки текстом. Значение 0 соответствовало полному противоречию, а значение 10 — полной поддержке. Формулировка задания для модели приведена в Приложении А.4.

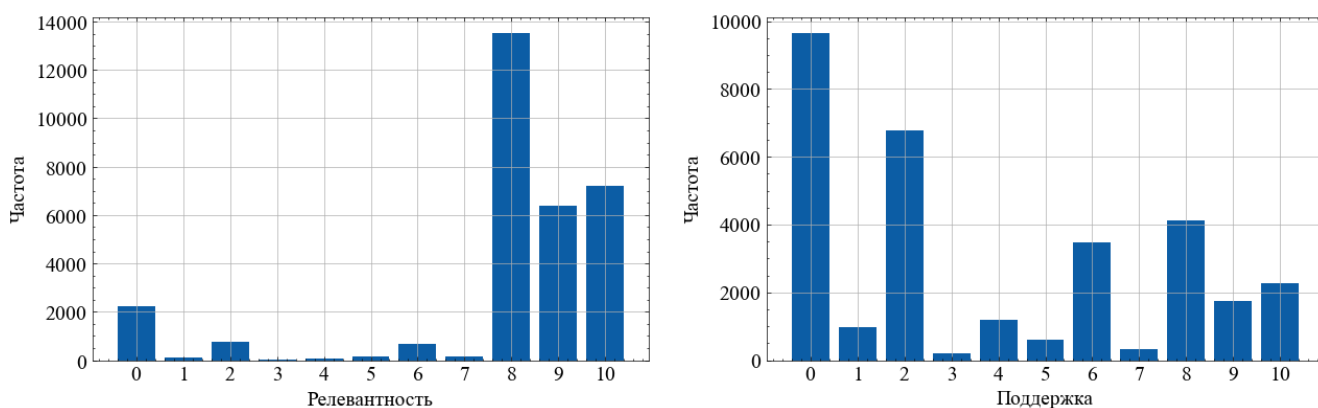


Рисунок 4.2 — Распределение оценок релевантности (слева) и поддержки (справа), полученных от языковой модели при генерации противоречащих утверждений.

Распределения полученных оценок представлены на Рисунке 4.2. Гистограммы показывают, что модель часто порождала либо высокорелевантные утверждения с оценками 8–10, либо полностью нерелевантные утверждения с оценкой 0. Распределение оценок поддержки является более равномерным, но также имеет пики на крайних значениях. Это означает, что среди сгенерированных примеров встречались как явные противоречия с оценкой 0, так и ошибочные подтверждения.

В итоговую автоматическую выборку включались только пары с релевантностью не ниже 7 и степенью поддержки не выше 4. Пороговые значения были выбраны эмпирически по результатам экспертной разметки контрольной подвыборки. Это позволило приблизить автоматический отбор к человеческой оценке и исключить значительную часть нерелевантных или некорректных отрицательных примеров.

4.4 Экспертная валидация и характеристики набора данных

Даже после многоэтапной автоматической фильтрации в сгенерированных данных могли сохраняться артефакты, смысловые неточности и неоднозначные случаи. Для их выявления был проведен заключительный этап ручной проверки корпуса.

К работе были привлечены два ассессора-терминолога, обладающие опытом анализа научных текстов в различных предметных областях. Каждой паре (c, e) необходимо было присвоить одну из трех меток: подтверждает, опровергает или проблемный. Данная метка предназначалась для случаев, когда утверждение было сформулировано неоднозначно, содержало несуществующие термины, его истинность нельзя было установить на основе аннотации или оно имело другие дефекты, делающие его непригодным для оценки.

Для контроля качества разметки была проведена перекрестная проверка: случайная подвыборка из 300 примеров была независимо размечена двумя экспертами. Согласованность оценивалась с помощью коэффициента Коэна [76], который учитывает не только наблюдаемую долю совпадений, но и совпадения, ожидаемые при случайной разметке:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \quad p_e = \sum_{i=1}^k p_i p_{+i} = \frac{1}{N^2} \sum_{i=1}^k n_{i+} n_{+i},$$

где p_o - наблюдаемая доля совпадений между разметчиками, p_e - ожидаемая доля случайных совпадений, n_{i+} и n_{+i} - числа примеров, отнесенных к классу i первым и вторым разметчиком соответственно, N - размер контрольной подвыборки. Полученное значение $\kappa = 0,86$ свидетельствует о высокой согласованности экспертной разметки. После согласования критериев оставшиеся 828 пар были размечены одним экспертом.

В результате описанной процедуры был сформирован итоговый набор данных RuSciFact, включающий 1128 пар «утверждение–аннотация». Детальный анализ его характеристик позволяет оценить его сложность и репрезентативность.

Статистика длин текстов приведена в Таблице 19. Утверждения в наборе данных являются короткими и атомарными: их медианная длина составляет 14 слов. Аннотации заметно длиннее, их медианная длина равна 144 словам. Такое соотношение отражает постановку задачи, в которой для проверки краткого

утверждения необходимо сопоставить его с более развернутым научным контекстом.

Таблица 19 — Статистические характеристики длин текстов в наборе данных RuSciFact

| Тип текста | Среднее | 25% квантиль | 50% квантиль | 75% квантиль |
|-------------|---------|--------------|--------------|--------------|
| Утверждение | 15 | 12 | 14 | 18 |
| Аннотация | 165 | 86 | 144 | 234 |

Распределение классов представлено в Таблице 20. В наборе данных преобладают подтверждающие примеры: их 758 против 370 опровергающих. Таким образом, соотношение классов составляет приблизительно 2:1.

Таблица 20 — Распределение классов в итоговом наборе данных RuSciFact

| Метка | Количество |
|--------------|------------|
| подтверждает | 758 |
| опровергает | 370 |
| Всего | 1128 |

Тематический состав бенчмарка определялся по рубрикатору ГРНТИ исходных публикаций и приведен в Таблице 21. Набор данных охватывает разные научные дисциплины. Наиболее крупная категория — «Медицина и здравоохранение» (30.85%), что соответствует распределению публикаций в исходном корпусе RuSciBench. При этом в наборе представлены и другие области, включая физику, биологию, химию и инженерные науки, что обеспечивает тематическое разнообразие данных.

4.5 Экспериментальная оценка и анализ результатов

Созданный набор данных RuSciFact был использован для проведения комплексной оценки широкого спектра современных моделей семантического представления текстов. Эксперименты проводились в соответствии с двумя подзадачами, определенными в Разделе 4.1: информационный поиск и классификация. Цель исследования состояла в том, чтобы определить, насколько

Таблица 21 — Тематическое распределение данных RuSciFact по областям науки (рубрикатор ГРНТИ, 1-й уровень, представлены категории с долей >1%)

| Научная область (ГРНТИ) | Доля (%) |
|--|----------|
| Медицина и здравоохранение | 30.85 |
| Физика | 7.09 |
| Биология | 6.83 |
| Химия | 6.12 |
| Сельское и лесное хозяйство | 5.85 |
| Машиностроение | 3.99 |
| Механика | 3.46 |
| Геология | 3.28 |
| Полиграфия. Репрография. Фотокинетика | 3.10 |
| Математика | 2.75 |
| Народное образование. Педагогика | 2.04 |
| Строительство. Архитектура | 1.95 |
| Горное дело | 1.68 |
| Автоматика. Вычислительная техника | 1.68 |
| Языкознание | 1.51 |
| Психология | 1.51 |
| Государство и право. Юридические науки | 1.33 |
| Электроника. Радиотехника | 1.24 |
| Электротехника | 1.15 |
| Информатика | 1.15 |
| Металлургия | 1.06 |
| Остальные | 8.43 |

эффективно существующие векторные представления справляются с задачами тонкого семантического анализа научных текстов, требующими логического вывода, а также позиционировать разработанные в рамках диссертации модели семейства SciRus в контексте современных решений.

4.5.1 Оценка в задаче информационного поиска

Первая подзадача состояла в поиске единственно верной аннотации-источника для каждого утверждения из набора данных. В качестве основной меры качества использовалась метрика $MRR@1$ (Mean Reciprocal Rank at 1), которая показывает долю случаев, когда релевантная аннотация была ранжирована на первое место. Высокое значение этой меры свидетельствует о способности модели точно сопоставлять семантически близкие, но лексически различные тексты.

Результаты эксперимента представлены в Таблице 22. Анализ полученных данных позволяет сделать ряд ключевых выводов.

- **Преимущество крупномасштабных моделей.** Верхние строчки рейтинга занимают модели с миллиардами параметров, включая GritLM-7B, SFR-Embedding-2_R и multilingual-e5-large-instruct. Их результаты достигают 0.93–0.95 по мере $MRR@1$. Это показывает, что в задаче поиска релевантной аннотации по сложному утверждению масштаб модели и разнообразие обучающих данных существенно влияют на качество. Такие модели формируют векторное пространство, в котором лучше сохраняются тонкие семантические связи между утверждениями и аннотациями.
- **Эффективность компактных моделей.** Некоторые компактные модели также показывают высокие результаты. В частности, BERTA (128 млн параметров) достигает значения 0.92, а rubert-mini-frida (32 млн параметров) — 0.88, опережая ряд более крупных моделей. Это указывает на то, что архитектура, состав обучающих данных и режим обучения могут частично компенсировать меньший размер модели.
- **Позиционирование моделей SciRus.** Модели семейства SciRus показывают конкурентоспособные результаты среди компактных решений. Их значения $MRR@1$ находятся в диапазоне 0.70–0.75, что выше результатов ряда общецелевых многоязычных моделей сопоставимого или большего размера, включая rubert-tiny-turbo и LaBSE-en-ru. При этом они уступают лидирующим крупным моделям. Следовательно, адаптация к научным текстам дает преимущество над рядом базовых моделей, но в задаче точного поиска релевантного источника масштаб модели остается важным фактором.

- **Ограниченность части подходов.** Некоторые модели, особенно более ранние архитектуры, показывают результаты, близкие к случайным. Например, `rubert-tiny` достигает значения 0.09. Это подтверждает сложность задачи и показывает, что для ее решения требуется векторное пространство, хорошо отражающее смысловые отношения между научными утверждениями и аннотациями.

Таблица 22 — Результаты оценки моделей-кодировщиков в задаче информационного поиска на RuSciFact

| Название модели | Количество параметров | MRR@1 |
|---------------------------------------|-----------------------|-------|
| GritLM-7B | 7.24 млрд | 0.95 |
| SFR-Embedding-2_R | 7.11 млрд | 0.94 |
| gte-Qwen2-1.5B-instruct | 1 млрд | 0.93 |
| Linq-Embed-Mistral | 7 млрд | 0.93 |
| SFR-Embedding-Mistral | 7 млрд | 0.93 |
| multilingual-e5-large-instruct | 560 млн | 0.93 |
| FRIDA | 823 млн | 0.92 |
| BERTA | 128 млн | 0.92 |
| jina-embeddings-v3 | 572 млн | 0.92 |
| gte-Qwen2-7B-instruct | 7 млрд | 0.88 |
| multilingual-e5-base | 278 млн | 0.88 |
| rubert-mini-frida | 32 млн | 0.88 |
| multilingual-e5-large | 560 млн | 0.86 |
| USER-bge-m3 | 359 млн | 0.85 |
| multilingual-e5-small | 118 млн | 0.80 |
| SciRus-small-cite | 61 млн | 0.75 |
| USER-base | 124 млн | 0.73 |
| SciRus-tiny-cite | 23 млн | 0.70 |
| SciRus-small | 61 млн | 0.70 |
| SciRus-tiny | 23 млн | 0.71 |
| rubert-tiny-turbo | 29 млн | 0.66 |
| LaBSE-en-ru | 129 млн | 0.53 |
| paraphrase-multilingual-mpnet-base-v2 | 278 млн | 0.49 |

Продолжение таблицы 22

| Название модели | Количество параметров | MRR@1 |
|---|-----------------------|-------|
| GIST-large-Embedding-v0 | 335 млн | 0.24 |
| rubert-tiny2 | 29 млн | 0.22 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 278 млн | 0.17 |
| rubert-tiny | 12 млн | 0.09 |

4.5.2 Оценка в задаче классификации

Вторая подзадача заключалась в классификации предоставленной пары «утверждение–аннотация» на два класса: подтверждает или опровергает. В отличие от поиска, эта задача требует не только установления тематической близости, но и проведения логического вывода для определения характера взаимосвязи. В качестве меры качества использовалась F1-мера.

Результаты, представленные в Таблице 23, показывают иную картину.

- **Более высокая сложность задачи.** Лидирующие позиции вновь занимают крупные модели, однако максимальное значение F1-меры (0.87 у gte-Qwen2-7B-instruct) заметно ниже, чем значения MRR@1 в задаче поиска. Это прямо указывает на то, что задача классификации, требующая логического вывода, является существенно более сложной для современных векторных представлений, чем задача поиска по семантической близости.
- **Уплотнение результатов.** Ключевой особенностью является сильное уплотнение результатов в середине таблицы. Широкий спектр моделей с разным числом параметров и архитектурой (от 23 млн до 823 млн) показывает очень близкие F1-оценки в диапазоне 0.67–0.70. В эту группу попадают и модели семейства SciRus, и многоязычные модели, и общие модели для русского языка. Это явление можно интерпретировать как наличие «плато производительности»: многие современные кодировщики способны уловить общую тематическую связь между утверждением и ан-

нотацией, но им не хватает разрешающей способности для надежного различения подтверждения и тонкого семантического противоречия.

- **Отсутствие явной корреляции с размером.** В отличие от задачи поиска, в задаче классификации преимущество крупных моделей не столь выражено. Например, GritLM-7B (7.24 млрд параметров), лидер задачи поиска, здесь показывает результат 0.73, уступая гораздо более компактным моделям. Это подтверждает, что для задач, требующих логического вывода, простое увеличение масштаба модели не всегда приводит к пропорциональному росту качества.

В целом, проведенный анализ показывает, что созданный бенчмарк RuSciFact является эффективным инструментом для дифференцированной оценки языковых моделей. Он выявляет, что если задача поиска релевантного научного контекста для современных крупных моделей близка к решению, то задача логической верификации этого контекста остается открытым вызовом, указывая на необходимость разработки новых архитектур и методов обучения, нацеленных на улучшение способностей моделей к логическому выводу.

Таблица 23 — Результаты оценки моделей-эмбеддеров в задаче классификации на RuSciFact

| Название модели | Количество параметров | F1 |
|--------------------------------|-----------------------|------|
| gte-Qwen2-7B-instruct | 7 млрд | 0.87 |
| SFR-Embedding-2_R | 7.11 млрд | 0.82 |
| Linq-Embed-Mistral | 7 млрд | 0.81 |
| SFR-Embedding-Mistral | 7 млрд | 0.80 |
| multilingual-e5-large-instruct | 560 млн | 0.77 |
| gte-Qwen2-1.5B-instruct | 1 млрд | 0.74 |
| GritLM-7B | 7.24 млрд | 0.73 |
| USER-base | 124 млн | 0.70 |
| BERTA | 128 млн | 0.68 |
| USER-bge-m3 | 359 млн | 0.68 |
| rubert-tiny-turbo | 29 млн | 0.68 |
| SciRus-small-cite | 61 млн | 0.68 |
| SciRus-small | 61 млн | 0.68 |
| multilingual-e5-large | 560 млн | 0.68 |

Продолжение таблицы 23

| Название модели | Количество параметров | F1 |
|---|-----------------------|------|
| LaBSE-en-ru | 129 млн | 0.68 |
| FRIDA | 823 млн | 0.67 |
| rubert-tiny | 12 млн | 0.67 |
| paraphrase-multilingual-mpnet-base-v2 | 278 млн | 0.67 |
| jina-embeddings-v3 | 572 млн | 0.67 |
| rubert-tiny2 | 29 млн | 0.67 |
| SciRus-tiny-cite | 23 млн | 0.67 |
| multilingual-e5-base | 278 млн | 0.67 |
| rubert-mini-frida | 32 млн | 0.67 |
| SciRus-tiny | 23 млн | 0.67 |
| multilingual-e5-small | 118 млн | 0.66 |
| sn-xlm-roberta-base-snli-mnli-anli-xnli | 278 млн | 0.63 |
| GIST-large-Embedding-v0 | 335 млн | 0.58 |

4.6 Основные выводы

В главе предложен новый бенчмарк RuSciFact для верификации научных фактов на русском языке. Задача формализована как декомпозиция на информационный поиск релевантной аннотации (4.1) и последующую бинарную классификацию пары «утверждение–аннотация» (4.2). Такая постановка позволяет отдельно оценивать способность моделей к семантическому сопоставлению и логическому выводу.

Основу бенчмарка составляет набор из 1128 пар «утверждение–аннотация», сформированный с помощью многоэтапной процедуры формирования набора данных. Утверждения генерировались большой языковой моделью с учетом семантических и логических требований. Для подтверждающих примеров учитывались логическая выводимость, неявность и обоснованность утверждения, а для противоречащих примеров — смысловое противоречие без прямого отрица-

ния. Качество данных повышалось за счет автоматической фильтрации, включая удаление тривиальных перефразирований по лексическому сходству, отбор более сложных примеров и применение схемы «LLM как судья» для оценки релевантности и степени поддержки. Надежность экспертной валидации подтверждается перекрестной разметкой контрольной подвыборки из 300 примеров, для которой значение Cohen's Kappa составило 0.86.

Экспериментальная оценка выявила принципиальное различие в сложности подзадач. В задаче поиска на наборе RuSciFact современные крупномасштабные модели демонстрируют высокие значения $MRR@1$, близкие к насыщению, что указывает на относительную простоту поиска релевантной аннотации в рамках данного тестового корпуса. В то же время в задаче классификации пары «утверждение–аннотация» по F1-мере наблюдается «плато производительности»: широкий спектр моделей показывает близкие и заметно более низкие результаты. Это свидетельствует о том, что логическое сопоставление утверждения с текстом аннотации остается более сложным этапом и является одним из ключевых ограничений существующих векторных представлений.

Таким образом, RuSciFact позволяет отдельно анализировать качество поиска релевантного научного контекста и качество логического сопоставления утверждения с этим контекстом. Полученные результаты показывают, что дальнейшее развитие моделей для научных текстов должно учитывать не только качество семантического поиска, но и способность к проверке отношений поддержки и противоречия.

Заключение

Основные результаты работы заключаются в следующем.

1. С помощью двухэтапной методологии обучения разработаны двуязычные модели для векторного представления научных текстов на русском и английском языках. Показана применимость использования пар «заголовков-аннотация» на этапе контрастивного обучения. Эксперименты на бенчмарках RuSciBench и SciDocs показали, что предложенные модели, несмотря на значительно меньшее число параметров, демонстрируют качество, сопоставимое с гораздо более крупными моделями, и обладают высокой вычислительной эффективностью.
2. Разработан и апробирован мультизадачный русско-английский бенчмарк RuSciBench, предназначенный для оценки качества моделей векторного представления научных текстов. Бенчмарк суммарно включает 18 задач, 9 из которых были разработаны лично, и основан на данных российской научной электронной библиотеки eLibrary.ru. RuSciBench обеспечивает стандартизированную и воспроизводимую процедуру тестирования и интегрирован в международный лидерборд MTEB. На основе данного бенчмарка впервые проведено масштабное сравнительное исследование широкого спектра современных моделей векторизации на задачах, связанных с научными текстами. Симметричная двуязычная структура бенчмарка позволила количественно оценить влияние языка задачи на производительность моделей и выявить степень языковой специализации каждой из них.
3. Предложена и экспериментально проверена полуавтоматическая методика формирования наборов данных для задачи верификации научных фактов на русском языке. Данная методика сочетает генерацию научных утверждений на основе аннотаций с использованием больших языковых моделей (LLM), многоэтапную фильтрацию и оценку сгенерированных утверждений самой моделью, а также последующую экспертную валидацию. На основе этой методики создан и опубликован RuSciFact - первый русскоязычный бенчмарк для оценки способности моделей определять, подтверждается ли научное утверждение текстом аннотации или проти-

воречит ему. Проведена оценка современных моделей векторизации на данном бенчмарке.

Список литературы

1. *Thelwall, M.* Scopus 1900–2020: Growth in articles, abstracts, countries, fields, and journals [Текст] / М. Thelwall, P. Sud // *Quantitative Science Studies*. — 2022. — Апр. — Т. 3, № 1. — С. 37–50. — eprint: https://direct.mit.edu/qss/article-pdf/3/1/37/2008360/qss_a_00177.pdf. — URL: https://doi.org/10.1162/qss%5C_a%5C_00177.
2. Динамика роста числа публикаций [Текст]. — ООО Научная электронная библиотека, 2025. — URL: https://www.elibrary.ru/stat_time_items.asp (дата обр. 10.05.2025) ; Статистические данные по состоянию на 04.05.2025. Электронный ресурс eLibrary.ru.
3. *Бажанов, Д. И.* Параллельная обработка данных в научных исследованиях [Текст] / Д. И. Бажанов, А. А. Журавлев, К. К. Абгарян. — Москва : МАКС ПРЕСС, 2021. — С. 112. — Тираж 500(1-50) экз.
4. *Salton, G.* A vector space model for automatic indexing [Текст] / G. Salton, A. Wong, C.-S. Yang // *Communications of the ACM*. — 1975. — Т. 18, № 11. — С. 613–620.
5. *Robertson, S.* The probabilistic relevance framework: BM25 and beyond [Текст] / S. Robertson, H. Zaragoza // *Foundations and Trends® in Information Retrieval*. — 2009. — Т. 3, № 4. — С. 333–389.
6. Attention is all you need [Текст] / A. Vaswani [и др.] // *Advances in neural information processing systems*. — Curran Associates, Inc. 2017. — С. 5998–6008.
7. BioCPT: Contrastive Pre-trained Transformers with Large-scale PubMed Search Logs for Zero-shot Biomedical Information Retrieval [Текст] / Q. Jin [и др.] // *Bioinformatics*. — 2023. — Т. 39 11. — URL: <https://api.semanticscholar.org/CorpusID:259316759>.
8. Text Embeddings by Weakly-Supervised Contrastive Pre-training [Текст] / L. Wang [и др.] // arXiv preprint arXiv:2212.03533. — 2022.
9. SciRus: Tiny and Powerful Multilingual Encoder for Scientific Texts [Текст] / N. Gerasimenko [и др.] // *Doklady Mathematics*. — 2024. — Дек. — Т. 110, № 1. — S193–S202. — URL: <https://doi.org/10.1134/S1064562424602178>.

10. RuSentEval: Linguistic Source, Encoder Force! [Текст] / V. Mikhailov [и др.] // Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing. — Kiyv, Ukraine : Association for Computational Linguistics, 04.2021. — С. 43—65. — URL: <https://aclanthology.org/2021.bsnlp-1.6>.
11. Dale, D. Рейтинг русскоязычных энкодеров предложений [Текст] / D. Dale. — 06.2022. — URL: <https://habr.com/ru/articles/669674/> ; [Online; posted 12-June-2022].
12. *Президент Российской Федерации*. Национальная стратегия развития искусственного интеллекта на период до 2030 года (с изменениями 2024 г.) [Текст] / Президент Российской Федерации. — 2024. — Утверждена Указом Президента Российской Федерации от 10 октября 2019 г. № 490. Изменения внесены Указом Президента Российской Федерации от 15 февраля 2024 г. № 124 "О внесении изменений в Указ Президента Российской Федерации от 10 октября 2019 г. N 490 'О развитии искусственного интеллекта в Российской Федерации' и в Национальную стратегию, утвержденную этим Указом".
13. Spärck Jones, K. A statistical interpretation of term specificity and its application in retrieval [Текст] / K. Spärck Jones // Journal of Documentation. — 1972. — Т. 28, № 1. — С. 11—21.
14. Indexing by latent semantic analysis [Текст] / S. Deerwester [и др.] // Journal of the American Society for Information Science. — 1990. — Т. 41, № 6. — С. 391—407.
15. Efficient estimation of word representations in vector space [Текст] / T. Mikolov [и др.] // arXiv preprint arXiv:1301.3781. — 2013.
16. Pennington, J. GloVe: Global Vectors for Word Representation [Текст] / J. Pennington, R. Socher, C. D. Manning // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Association for Computational Linguistics, 2014. — С. 1532—1543. — URL: <https://aclanthology.org/D14-1162>.
17. Deep contextualized word representations [Текст] / M. E. Peters [и др.] // arXiv preprint arXiv:1802.05365. — 2018.

18. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Текст] / J. Devlin [и др.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) / под ред. J. Burstein, C. Doran, T. Solorio. — Minneapolis, Minnesota : Association for Computational Linguistics, 06.2019. — С. 4171—4186. — URL: <https://aclanthology.org/N19-1423/>.
19. Reimers, N. Sentence-bert: Sentence embeddings using siamese bert-networks [Текст] / N. Reimers, I. Gurevych // arXiv preprint arXiv:1908.10084. — 2019.
20. Oord, A. van den. Representation Learning with Contrastive Predictive Coding [Текст] / A. van den Oord, Y. Li, O. Vinyals // arXiv preprint arXiv:1807.03748. — 2018. — Июль. — URL: <https://arxiv.org/abs/1807.03748>.
21. Beltagy, I. SciBERT: A Pretrained Language Model for Scientific Text [Текст] / I. Beltagy, K. Lo, A. Cohan // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). — Association for Computational Linguistics, 2019. — С. 3613—3618.
22. SPECTER: Document-level Representation Learning using Citation-informed Transformers [Текст] / A. Cohan [и др.] // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics / под ред. D. Jurafsky [и др.]. — Online : Association for Computational Linguistics, 07.2020. — С. 2270—2282. — URL: <https://aclanthology.org/2020.acl-main.207/>.
23. SciRepEval: A Multi-Format Benchmark for Scientific Document Representations [Текст] / A. Singh [и др.] // ArXiv. — 2022. — Т. abs/2211.13308.
24. Neighborhood Contrastive Learning for Scientific Document Representations with Citation Embeddings [Текст] / M. Ostendorff [и др.] // Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing / под ред. Y. Goldberg, Z. Kozareva, Y. Zhang. — Abu Dhabi, United Arab Emirates : Association for Computational Linguistics, 12.2022. — URL: <https://aclanthology.org/2022.emnlp-main.802/>.
25. MTEB: Massive Text Embedding Benchmark [Текст] / N. Muennighoff [и др.] // arXiv preprint arXiv:2210.07316. — 2022. — URL: <https://arxiv.org/abs/2210.07316>.

26. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models [Текст] / N. Thakur [и др.] // Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2). — 2021. — URL: <https://openreview.net/forum?id=wCu6T5xFjeJ>.
27. The Russian-focused embedders' exploration: ruMTEB benchmark and Russian embedding model design [Текст] / A. Snegirev [и др.] // arXiv preprint arXiv:2408.12503. — 2024. — arXiv: [2408.12503](https://arxiv.org/abs/2408.12503) [cs.CL].
28. Fact or Fiction: Verifying Scientific Claims [Текст] / D. Wadden [и др.] // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Online : Association for Computational Linguistics, 11.2020. — С. 7534—7550. — URL: <https://aclanthology.org/2020.emnlp-main.609>.
29. Zero-shot Fact Verification by Claim Generation [Текст] / L. Pan [и др.] // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) / под ред. С. Zong [и др.]. — Online : Association for Computational Linguistics, 08.2021. — С. 476—483. — URL: <https://aclanthology.org/2021.acl-short.61>.
30. Generating Biomedical Assertions from Text [Текст] / A. Wright [и др.] // Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. — 2022. — С. 4789—4800.
31. Unknown Claims: Generation of Fact-Checking Training Examples from Unstructured and Structured Data [Текст] / J.-F. Bussotti [и др.] // Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing / под ред. Y. Al-Onaizan, M. Bansal, Y.-N. Chen. — Miami, Florida, USA : Association for Computational Linguistics, 11.2024. — С. 12105—12122. — URL: <https://aclanthology.org/2024.emnlp-main.675/>.
32. *Kozlova, A.* Fact-checking benchmark for the Russian Large Language Models [Текст] / A. Kozlova, V. Mikhailov, T. Shavrina // INTERNATIONAL CONFERENCE on Computational Linguistics and Intellectual Technologies. — 2023.

33. *Kozlova, E.* Building a RuSSA Dataset for Contradiction Detection [Текст] / E. Kozlova, T. Shavrina, A. Fenogenova // Artificial Intelligence and Natural Language. AINL 2023. — Springer, 2023.
34. AIR-Bench: Automated Heterogeneous Information Retrieval Benchmark [Текст] / J. Chen [и др.] // Annual Meeting of the Association for Computational Linguistics. — 2024. — URL: <https://api.semanticscholar.org/CorpusID:274788729>.
35. Long Input Benchmark for Russian Analysis [Текст] / I. Churin [и др.] // ArXiv. — 2024. — Т. abs/2408.02439. — URL: <https://api.semanticscholar.org/CorpusID:271710181>.
36. RuSciBench: Open Benchmark for Russian and English Scientific Document Representations [Текст] / A. Vatolin [и др.] // Doklady Mathematics. — 2024. — Дек. — Т. 110, № 1. — S251—S260. — URL: <https://doi.org/10.1134/S1064562424602191>.
37. ruSciFact: Open Benchmark for Verifying Scientific Facts in Russian [Текст] / A. Vatolin [и др.] // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue" (2025). — 2025. — ИЮНЬ. — Т. 23. — S435—S459. — URL: <https://doi.org/10.28995/2075-7182-2025-23-435-459>.
38. MMTEB: Massive Multilingual Text Embedding Benchmark [Текст] / K. Enevoldsen [и др.] // International Conference on Learning Representations. Т. 2025 / под ред. Y. Yue [и др.]. — 2025. — С. 101715—101771. — URL: https://proceedings.iclr.cc/paper_files/paper/2025/file/fc0e3f908a2116ba529ad0a1530a3675-Paper-Conference.pdf.
39. *Vatolin, A.* Structured Sentiment Analysis with Large Language Models: A Winning Solution for RuOpinionNE-2024 [Текст] / A. Vatolin // Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue" (2025). — 2025. — ИЮНЬ. — Т. 23. — S416—S434. — URL: <https://doi.org/10.28995/2075-7182-2025-23-416-434>.
40. *Liu, Z.* Representation Learning for Natural Language Processing [Текст] / Z. Liu, Y. Lin, M. Sun // Representation Learning for Natural Language Processing. — 2023. — URL: <https://api.semanticscholar.org/CorpusID:63458344>.

41. *Firth, J. R.* A synopsis of linguistic theory 1930-1955 [Текст] / J. R. Firth // Studies in Linguistic Analysis, Special volume of the Philological Society. — 1957. — С. 1—32.
42. A Neural Probabilistic Language Model [Текст] / Y. Bengio [и др.] // Journal of machine learning research. — 2003. — Т. 3. — С. 1137—1155.
43. *Sennrich, R.* Neural machine translation of rare words with subword units [Текст] / R. Sennrich, B. Haddow, A. Birch // arXiv preprint arXiv:1508.07909. — 2015.
44. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation [Текст] / Y. Wu [и др.] // ArXiv. — 2016. — Т. abs/1609.08144. — URL: <https://api.semanticscholar.org/CorpusID:3603249>.
45. *Kudo, T.* Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates [Текст] / T. Kudo // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) / под ред. I. Gurevych, Y. Miyao. — Melbourne, Australia : Association for Computational Linguistics, 07.2018. — С. 66—75. — URL: <https://aclanthology.org/P18-1007/>.
46. A Robustly Optimized BERT Pre-training Approach with Post-training [Текст] / L. Zhuang [и др.] // Proceedings of the 20th Chinese National Conference on Computational Linguistics / под ред. S. Li [и др.]. — Huhhot, China : Chinese Information Processing Society of China, 08.2021. — С. 1218—1227. — URL: <https://aclanthology.org/2021.ccl-1.108/>.
47. Language Models are Unsupervised Multitask Learners [Текст] / A. Radford [и др.] // . — 2019. — URL: <https://api.semanticscholar.org/CorpusID:160025533>.
48. *Nair, V.* Rectified Linear Units Improve Restricted Boltzmann Machines [Текст] / V. Nair, G. E. Hinton // Proceedings of the 27th International Conference on Machine Learning. — 2010. — С. 807—814.
49. *Ba, J.* Layer Normalization [Текст] / J. Ba, J. R. Kiros, G. E. Hinton // ArXiv. — 2016. — Т. abs/1607.06450. — URL: <https://api.semanticscholar.org/CorpusID:8236317>.
50. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation [Текст] / Y. Wu [и др.] // ArXiv. — 2016. — Т. abs/1609.08144. — URL: <https://api.semanticscholar.org/CorpusID:3603249>.

51. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation [Текст] / D. Cer [и др.] // Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017) / под ред. S. Bethard [и др.]. — Vancouver, Canada : Association for Computational Linguistics, 08.2017. — С. 1—14. — URL: <https://aclanthology.org/S17-2001/>.
52. *Schroff, F.* FaceNet: A unified embedding for face recognition and clustering [Текст] / F. Schroff, D. Kalenichenko, J. Philbin // 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2015. — С. 815—823. — URL: <https://api.semanticscholar.org/CorpusID:206592766>.
53. *Koponen, I. T.* Usage of Terms "Science" and "Scientific Knowledge" in Nature of Science (NOS): Do Their Lexicons in Different Accounts Indicate Shared Conceptions? [Текст] / I. T. Koponen // Education Sciences. — 2020. — URL: <https://api.semanticscholar.org/CorpusID:225026010>.
54. *Pan, S. J.* A Survey on Transfer Learning [Текст] / S. J. Pan, Q. Yang // IEEE Transactions on Knowledge and Data Engineering. — 2010. — Т. 22, № 10. — С. 1345—1359.
55. Parameter-efficient transfer learning for NLP [Текст] / N. Houlsby [и др.] // International conference on machine learning. — PMLR. 2019. — С. 2790—2799.
56. *Pasternack, S.* The Scientific Enterprise: Public Knowledge. An Essay Concerning the Social Dimension of Science [Текст] / S. Pasternack // Science. — 1969. — Май. — Т. 164, № 3880. — С. 669—670. — URL: <https://www.science.org/doi/10.1126/science.164.3880.669>.
57. PyTorch-BigGraph: A Large-scale Graph Embedding System [Текст] / A. Lerer [и др.] // Proceedings of the 2nd SysML Conference. — Palo Alto, CA, USA, 2019.
58. *Lipscomb, C. E.* Medical subject headings (MeSH) [Текст] / C. E. Lipscomb // Bulletin of the Medical Library Association. — 2000. — Т. 88, № 3. — С. 265.
59. An overview of microsoft academic service (mas) and applications [Текст] / A. Sinha [и др.] // Proceedings of the 24th international conference on world wide web. — 2015. — С. 243—246.

60. S2ORC: The Semantic Scholar Open Research Corpus [Текст] / К. Ло [и др.] // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics / под ред. D. Jurafsky [и др.]. — Association for Computational Linguistics, 07.2020. — С. 4969—4983. — URL: <https://aclanthology.org/2020.acl-main.447>.
61. *Dale, D.* Маленький и быстрый BERT для русского языка [Текст] / D. Dale. — 06.2021. — URL: <https://habr.com/ru/post/562064/> ; [Online; posted 10-June-2021].
62. *Hugging Face.* Text Embeddings Inference [Текст] / Hugging Face. — 2023. — URL: <https://github.com/huggingface/text-embeddings-inference> ; Accessed: 2025-05-14. <https://github.com/huggingface/text-embeddings-inference>.
63. *Grafana Labs.* k6 [Текст] / Grafana Labs. — 2017. — URL: <https://k6.io> ; A modern load testing tool for developers and testers. <https://k6.io>.
64. RuMedBench: A Russian Medical Language Understanding Benchmark [Текст] / P. Blinov [и др.] // Conference on Artificial Intelligence in Medicine in Europe. — 2022. — URL: <https://api.semanticscholar.org/CorpusID:246016223>.
65. *Stahl, P. M.* Lingua-py: The most accurate natural language detection library for Python [Текст] / P. M. Stahl. — 2024. — URL: <https://github.com/pemistahl/lingua-py> ; Suitable for short text and mixed-language text detection. <https://github.com/pemistahl/lingua-py>.
66. *Lopez, F.* langdetect: Language detection library for Python [Текст] / F. Lopez. — 2021. — Дата обращения: 07.08.2025. <https://github.com/fedelopez77/langdetect>.
67. Learning Word Vectors for 157 Languages [Текст] / E. Grave [и др.] // Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018). — 2018.
68. Scikit-learn: Machine Learning in Python [Текст] / F. Pedregosa [и др.] // Journal of Machine Learning Research. — 2011. — Т. 12. — С. 2825—2830.
69. What are the best systems? new perspectives on NLP benchmarking [Текст] / P. Colombo [и др.] // Proceedings of the 36th International Conference on Neural Information Processing Systems. — New Orleans, LA, USA : Curran Associates Inc., 2022. — (NIPS '22).

70. *Gilardi, F.* ChatGPT outperforms crowd workers for text-annotation tasks [Текст] / F. Gilardi, M. Alizadeh, M. Kubli // Proceedings of the National Academy of Sciences of the United States of America. — 2023. — Т. 120. — URL: <https://api.semanticscholar.org/CorpusID:257766307>.
71. LLMs in the Loop: Leveraging Large Language Model Annotations for Active Learning in Low-Resource Languages [Текст] / N. Kholodna [и др.] // ArXiv. — 2024. — Т. abs/2404.02261. — URL: <https://api.semanticscholar.org/CorpusID:268876095>.
72. MERA: A Comprehensive LLM Evaluation in Russian [Текст] / A. Fenogenova [и др.] // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) / под ред. L.-W. Ku, A. Martins, V. Srikumar. — Bangkok, Thailand : Association for Computational Linguistics, 08.2024. — С. 9920—9948. — URL: <https://aclanthology.org/2024.acl-long.534>.
73. Efficient Memory Management for Large Language Model Serving with PagedAttention [Текст] / W. Kwon [и др.] // Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles. — 2023.
74. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale [Текст] / T. Dettmers [и др.] // Advances in Neural Information Processing Systems. — 2022. — URL: <https://arxiv.org/abs/2208.07339>.
75. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena [Текст] / L. Zheng [и др.]. — 2023. — arXiv: [2306.05685](https://arxiv.org/abs/2306.05685) [cs.CL].
76. *Cohen, J.* A Coefficient of Agreement for Nominal Scales [Текст] / J. Cohen // Educational and Psychological Measurement. — 1960. — Т. 20, № 1. — С. 37—46. — URL: <https://doi.org/10.1177/001316446002000104>.

Приложение А

Промпты для формирования выборки в бенчмарке RuSciFact и примеры данных

А.1 Промпты для генерации подтверждающего утверждения в бенчмарке RuSciFact

Этот промпт генерирует один строгий факт, который следует из аннотации, без прямого цитирования. Плейсхолдер {text} заменяется текстом аннотации.

Вы ученый, который хорошо разбирается во всех областях науки. Ваша задача — сформулировать один факт, который следует из аннотации к статье. Вы не можете скопировать текст из аннотации, вам нужно написать факт, который следует из аннотации, но прямо в ней не указан. Примечание: убедитесь, что извлеченный факт точно выведен из содержания аннотации, без добавления какой-либо дополнительной информации или интерпретации. При написании факта избегайте ссылок на аннотацию (в приведенном тексте, в данной работе, предложенный метод). Также при написании факта избегайте неопределенности, например ”с определенными свойствами”, ”в некоторых состояниях”, ”определенной длины”. Если по аннотации невозможно написать точный факт, то напишите ”Аннотация не содержит факт”.

Ниже приведены 2 примера фактов и аннотаций, на основе которых они были написаны.

Аннотация к статье: Ожидается, что снижение уровня гомоцистеина в сыворотке крови с помощью фолиевой кислоты снизит смертность от ишемической болезни сердца. Известно, что максимальное снижение уровня гомоцистеина достигается при приеме фолиевой кислоты в дозе 1 мг/сут, но эффект более низких доз (имеющих отношение к обогащению пищевых продуктов) неясен. МЕТОДЫ Мы рандомизировали 151 пациента с ишеми-

ческой болезнью сердца на 1 из 5 доз фолиевой кислоты (0,2, 0,4, 0,6, 0,8 и 1,0 мг/сут) или плацебо. Первоначально, через 3 месяца приема добавок и через 3 месяца после прекращения приема фолиевой кислоты, были взяты образцы крови натощак для анализа на содержание гомоцистеина и фолиевой кислоты в сыворотке крови. РЕЗУЛЬТАТЫ: Средний уровень гомоцистеина в сыворотке крови снижался при увеличении дозы фолиевой кислоты до максимума при приеме 0,8 мг фолиевой кислоты в день, когда снижение уровня гомоцистеина (с поправкой на плацебо) составляло 2,7 мкмоль/л (23%), что аналогично известному эффекту приема фолиевой кислоты в дозах 1 мг/сут и выше. Чем выше был исходный уровень гомоцистеина в сыворотке крови человека, тем сильнее была реакция на фолиевую кислоту, но статистически значимое снижение наблюдалось независимо от исходного уровня. Уровень фолиевой кислоты в сыворотке крови повышался примерно линейно (5,5 нмоль/л на каждые 0,1 мг фолиевой кислоты). Индивидуальные колебания уровня гомоцистеина в сыворотке крови, измеренные в группе плацебо, были значительными по сравнению с эффектом приема фолиевой кислоты, что указывает на то, что мониторинг снижения уровня гомоцистеина у конкретного человека нецелесообразен. ВЫВОДЫ Для достижения максимального снижения уровня гомоцистеина в сыворотке крови во всем диапазоне уровней гомоцистеина в популяции, по-видимому, необходима доза фолиевой кислоты в размере 0,8 мг/сут. Нынешние уровни обогащения пищевых продуктов в США позволят достичь лишь небольшой доли достижимого снижения уровня гомоцистеина.

Факт из статьи: Прием фолиевой кислоты снижает уровень гомоцистеина в крови.

Аннотация к статье: Для реакции выделения кислорода (OER) были разработаны катализаторы на основе распространенных в земной коре переходных металлов первого ряда (3d); однако они работают при потенциалах, значительно превышающих термодинамические требования. Теория функционала плотности предполагает, что высоковалентные металлы, не относящиеся к 3d-ряду, такие как вольфрам, могут модифицировать оксиды

3d-металлов, обеспечивая почти оптимальную энергию адсорбции предполагаемых промежуточных продуктов. Мы разработали метод синтеза при комнатной температуре для получения гелеобразных оксигидроксидных материалов с атомарно однородным распределением металлов. Эти гелеобразные оксигидроксиды FeCoW обладают самым низким перенапряжением (191 милливольт), зарегистрированным при плотности тока 10 миллиампер на квадратный сантиметр в щелочном электролите. Катализатор не проявляет признаков разложения после более чем 500 часов работы. Рентгеновская абсорбционная спектроскопия и вычислительные исследования показывают синергетическое взаимодействие между вольфрамом, железом и кобальтом в создании благоприятной локальной координационной среды и электронной структуры, улучшающих энергетику реакции.

Факт из статьи: Усовершенствованные катализаторы OER демонстрируют стабильную активность в течение нескольких сотен часов.

Если в аннотации не содержится фактов, например: В статье рассмотрено становление британо-японских отношений в период биполярности, отражена история формирования двусторонних отношений, а также сотрудничество в экономической, политической, научно-технической и социально-культурной сферах в указанный период. Главный акцент был сделан на рассмотрении зарождения отношений между Великобританией и Японией и анализе динамики их развития. то напишите "Аннотация не содержит факт".

Теперь ваша задача — сформулировать факт по следующей аннотации, точно следуя инструкциям.

{text}

Не пишите никаких вводных слов (например, "из аннотации следует, что"), только факт. Сформулируйте факт по этой аннотации:

А.2 Промпт для классификации сложности факта

Этот промпт определяет уровень сложности уже сформулированного факта. На вход подставляется сам факт вместо {text}; на выходе требуется одно слово: простой, средний, сложный или неопределенный.

Инструкция по классификации фактов по научным статьям

Вы являетесь учёным, обладающим глубокими знаниями во всех областях науки. Ваша задача — определить сложность предоставленного факта. Факты могут быть классифицированы по трём уровням сложности: простой, средний и сложный. В отдельных случаях факт может быть неопределённым.

Категории фактов:

1. Простой факт:

Факт очевиден большинству образованных людей и не требует дополнительных исследований или чтений для его подтверждения или опровержения. Такие факты известны из общих знаний.

Примеры простых фактов:

- Экономическая эффективность потребления может варьироваться в зависимости от личностных свойств потребителей.
- В России существует закон, регулирующий проведение медицинских научных исследований с участием человека и/или лабораторных животных.
- Нитраты могут вызывать токсические эффекты у животных.
- Российские вузы сокращают количество бюджетных мест.
- В Республике Алтай представлены многочисленные виды туризма и отдыха.
- У студентов вузов ценность внутреннего успеха выше, чем внешнего.
- Изменение жесткости элементов конструкции здания может быть вызвано разными факторами.
- Преступления, связанные с фальшивомонетничеством, совершались на территории Российской Федерации и Белгородской области.

2. Средний факт:

Факт достаточно сложен, большинству людей понадобится читать специализированные статьи или проводить запросы в интернете, чтобы понять, подтвердить или опровергнуть этот факт.

Примеры средних фактов:

- Поражение молочной железы эхинококком может быть излечено хирургическим путем.
- Татарстан стал более засушливым регионом за последние десятилетия.

3. Сложный факт:

Факт требует специфических знаний или экспертизы в данной научной области для его понимания.

Примеры сложных фактов:

- Прокатка СВС-продуктов в валках прокатного стана перед измельчением в шаровой мельнице увеличивает эффективность измельчения.
- У больных диабетическим макулярным отеком наблюдается повышенный уровень брадикинина в крови.
- Стентирование коронарных артерий не вызывает значимых изменений показателей глобальной и сегментарной продольной систолической деформации миокарда левого желудочка в первые 3 сут после процедуры.

4. Неопределённый факт:

Факт недостаточно ясен, неполон или содержит отсылки, требующие дополнительных исследований.

Примеры неопределённых фактов:

- Российские компании могут использовать разработанную шкалу для определения уровня развития ориентации на бренд.
- Предыдущие модели теплоусвоения вермикулита были неточными.

Ваша задача:

Внимательно изучите предоставленный факт и напишите одно из следующих определений сложности факта: "простой" "средний" "сложный" или "неопределённый". Напишите только одно слово, не объясняйте причины такой классификации.

Вот факт, для которого это нужно написать: {text}

А.3 Промпт для генерации опровергающего факта

Этот промпт формирует релевантное аннотации утверждение, которое не вытекает из текста и не подтверждается им. Нельзя использовать явное отрицание, требуется конкретный, проверяемый факт. На вход подается аннотация вместо {text}.

Вы ученый, который хорошо разбирается во всех областях науки. Ваша задача — сформулировать один факт, который был бы релевантен аннотации, но не следовал бы из нее и противоречил бы ее содержанию. Не используйте прямое отрицание; вместо этого напишите конкретное утверждение, которое не подтверждается аннотацией, но относится к ее теме. Вы не можете скопировать текст из аннотации, вам нужно написать факт, который не следует из аннотации. Примечание: убедитесь, что утверждение является проверяемым и не добавляет бессмысленных или нерелевантных деталей. При написании факта избегайте ссылок на аннотацию (в приведенном тексте, в данной работе, предложенный метод). Также при написании факта избегайте неопределенности, например ”с определенными свойствами ”в некоторых состояниях ”определенной длины”. Если по аннотации невозможно написать точный факт, то напишите ”Аннотация не содержит факт”.

Ниже приведены 2 примера фактов и аннотаций, на основе которых они были написаны.

Пример 1

Аннотация к статье:

Статья посвящена актуальным проблемам установления уголовного запрета в сфере профессиональной медицинской деятельности. Проанализированы предложения Следственного комитета РФ о внесении изменений в действующий Уголовный Кодекс РФ, на основании социологического опроса и изучения уголовных дел, сделаны выводы о необходимости реформы

уголовного закона. Основным является вывод о невозможности решения актуальных проблем в российском здравоохранении исключительно уголовно-правовыми средствами.

Факт из статьи: Актуальные проблемы в российском здравоохранении могут быть решены исключительно уголовно-правовыми средствами

Пример 2

Аннотация к статье:

Рассмотрены вопросы создания системы охраны территорий и объектов стратегического назначения. Предложены структура и способ построения такой системы, использующие методы теории решеток. Для обработки и анализа информации с датчиков физических величин и последующего принятия решений применяются решетки, построенные с помощью оператора замыкания.

Факт из статьи: Решетки могут быть построены без использования оператора замыкания

Пример аннотации без фактов

Если в аннотации не содержится фактов, например:

В статье рассмотрено становление британо-японских отношений в период биполярности, отражена история формирования двусторонних отношений, а также сотрудничество в экономической, политической, научно-технической и социально-культурной сферах в указанный период. Главный акцент был сделан на рассмотрении зарождения отношений между Великобританией и Японией и анализе динамики их развития.

то напишите "Аннотация не содержит факт".

Пример неподходящего факта

Аннотация к статье: В статье обсуждаются вопросы взаимосвязи токсичности сточных вод и их химического состава. Для ряда гидрохимических показателей установлена достоверная связь между показателем токсичности, определявшимся с использованием методики, где в качестве тест-организма выступает *P. Caudatum*.

Факт из статьи: Для ряда гидрохимических показателей не существует достоверной связи между показателем токсичности,

определявшимся с использованием методики, где в качестве тест-организма выступает *P. caudatum*.

Факт не подходит, потому что можно удалить "не" и факт будет верным.

Не добавляйте ничего к ответу, напишите только факт. Не пишите свои рассуждения, только факт! Теперь ваша задача — сформулировать факт по следующей аннотации, точно следуя инструкциям:

Аннотация к статье: {text}

Факт из статьи:

А.4 Промпт для оценки релевантности и степени подтверждения

Этот промпт оценивает по шкале от 0 до 10 два аспекта: релевантность факта по отношению к аннотации и степень подтверждения факта текстом. На вход подставляется текст, содержащий аннотацию и соответствующий факт, через {text}; на выход требуется JSON с полями *relevance* и *support*.

Описание задачи

Вы ученый, который хорошо разбирается во всех областях науки.

Задача

Задача — оценить релевантность факта по отношению к аннотации и степень подтверждения факта аннотацией.

Формат вывода

На выходе нужно написать JSON.

Пример ответа:

```
{  
  "relevance": "Релевантность факта по отношению к аннотации",  
  "support": "Насколько аннотация подтверждает факт"  
}
```

Описание полей

Поля `relevance` и `support` могут принимать значения от 0 до 10. Для `relevance` значение 0 означает полную нерелевантность факта по отношению к аннотации, а 10 — максимальную релевантность. Для `support` значение 0 означает полное противоречие или отсутствие поддержки, а 10 — полное подтверждение факта аннотацией.

Входные данные

Текст для анализа:
{text}

A.5 Примеры положительных и отрицательных утверждений из RuSciFact

В этом разделе приведён один положительный и один отрицательный пример из датасета RuSciFact.

Пример 1

Утверждение: Пациенты с ишемической болезнью сердца, перенесшие чрескожные коронарные вмешательства, демонстрируют улучшение толерантности к физической нагрузке при использовании компьютеризированных систем поддержки врачебных решений.

Аннотация: Цель. Изучить эффективность амбулаторных реабилитационно-профилактических программ у пациентов после чрескожных коронарных вмешательств (ЧКВ) с использованием компьютеризированной системы поддержки врачебных решений (СПКР), предназначенной для выбора режима контролируемых физических тренировок (КФТ) и предоставления полноценных рекомендаций по физической активности (ФА). Материал и методы. Исследование выполняли в течение 12 мес. с включением 194 пациентов (124 мужчины и 70 женщин, средний возраст 53,5) со стабильной формой ишемической болезни сердца (ИБС), перенесших ЧКВ (коронарную ангиопластику, коронарное стентирование). При выборе режима КФТ использовалась компьютеризированная СПКР. Традиционные врачебные решения анализировали по специально разработанной анкете. Результаты. Пациенты группы КФТ, продемонстрировали достоверное увеличение толерантности к физической нагрузке (ТФН) и средней продолжительности ФН, положительную динамику качества жизни (КЖ), высокий уровень приверженности лекарственной терапии на протяжении всего периода реабилитации. При формировании врачебных решений использовали, в среднем, 3 клинических признака. Наиболее типичные врачебные ошибки носили методологический характер. Заключение. Интегрирование реабилитационных программ с использованием СПКР в амбулаторных условиях у пациентов, перенесших ЧКВ, обеспечивает высокую эффективность реабилитационно-профилактических мероприятий и безопасность ФТ.

Метка класса: подтверждает

Пример 2

Утверждение: Порода карпа не влияет на содержание сухого вещества, жира и протеина у сеголетков карпа.

Аннотация: В статье приведены результаты сравнения биохимического состава тела сеголетков и годовиков некоторых коллекционных пород карпа, разводимых в СПУ «Изобелино»: немецкого, сарбоянского, отводков изобелинского карпа (столин XVIII, три прим, смесь чешуйчатая), выращенных одновременно, в одинаковых условиях и зимовавших совместно в одном пруду. Установлены породы, характеризующиеся повышенными уровнями содержания сухого вещества, жира и протеина у сеголетков карпа. В результате исследования биохимического состава тела сеголетков карпа разной породной принадлежности, выращенных в одинаковых условиях, проявляется тенденция к увеличению содержания сухого вещества, жира и протеина у коллекционных линий карпа белорусской селек-

ции (изобелинский) по сравнению с породами зарубежной селекции (немецкий и сарбоянский), выращенными одновременно в одинаковых условиях. У годовиков коллекционных линий белорусской селекции отмечается тенденция к увеличению содержания сухого вещества, жира и протеина, снижению содержания влаги по сравнению с зимовавшими совместно коллекционными породами зарубежной селекции. В результате исследования изменения показателей, характеризующих биохимический состав тела, произошедших за зимний период, установлено, что отклонения биохимических показателей, особенно содержания сухого вещества и жира у пород зарубежной селекции, значительно выше, чем у линий изобелинского карпа (белорусская селекция). Полученные данные свидетельствуют о большей приспособленности карпа коллекционных линий белорусской селекции к условиям зимовки в Беларуси по сравнению с импортными породами (немецким и сарбоянским).

Метка класса: опровергает