УДК 004.8:004.912

**Д. Д. Чернышов**

*РТУ МИРЭА — Российский технологический университет, Москва, Россия*

**В. А. Романенко**

*РТУ МИРЭА — Российский технологический университет, Москва, Россия*

**Е. Е. Юдина**

*Московский физико-технический институт (национальный исследовательский университет), Долгопрудный, Россия*

# МНОГОУРОВНЕВАЯ ОЦЕНКА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ДОКАЗАТЕЛЬНОЙ ВЕРИФИКАЦИИ РУССКОЯЗЫЧНЫХ НОВОСТНЫХ ТЕКСТОВ

**Аннотация.** Представлена интеллектуальная система доказательной верификации русскоязычных новостных текстов, в которой итоговый вердикт формируется на основе сопоставления атомарных проверяемых утверждений с фрагментами доверенных публикаций. Система реализует последовательный конвейер лингвистической обработки, поиска и индексирования доказательств, распознавания логического следования и агрегации частных решений. Для адаптации семантического классификатора сформирован специализированный корпус из 391 пары «утверждение — доказательство», включающий контролируемые синтетические искажения и примеры ручной разметки по классам ENTAILS, CONTRADICTS и NEUTRAL. Предложена многоуровневая схема экспериментальной оценки, разделяющая качество NLI-модели, модуля верификации и полного конвейера. Дообученная модель RuBERT достигла Accuracy 0,878 и Macro-F1 0,875; в составе модуля верификации получены Accuracy 0,756 и Macro-F1 0,752. На сквозной выборке из 234 образцов система обеспечила полноту поиска Recall@10 0,761, MRR 0,612 и Accuracy 0,578 на образцах, для которых сформирован содержательный вердикт. Эксперимент с вычислительным окружением показал ускорение модуля верификации в 5,57 раза и сквозного серверного маршрута в 3,07 раза при использовании

графического процессора. Полученные результаты подтверждают эффективность декомпозиции автоматизированного фактчекинга на интерпретируемые стадии и позволяют количественно локализовать источники ошибок в каскадной системе.

**Ключевые слова:** автоматизированный фактчекинг, недостоверная информация, обработка естественного языка, поиск доказательств, распознавание логического следования, RuBERT, векторный поиск, объяснимый искусственный интеллект.

**Abstract.** The paper presents an intelligent evidence-based verification system for Russian-language news texts. The final verdict is derived by matching atomic checkable claims against evidence fragments retrieved from trusted publications. The system implements a sequential pipeline comprising linguistic processing, evidence retrieval and indexing, natural language inference, and aggregation of claim-level decisions. A domain-oriented dataset of 391 claim–evidence pairs was constructed to adapt the semantic classifier; it combines controlled synthetic distortions and manually annotated examples for the ENTAILS, CONTRADICTS, and NEUTRAL classes. A multi-level evaluation methodology is proposed to distinguish the quality of the NLI model, the verification service, and the end-to-end pipeline. The fine-tuned RuBERT model achieved an accuracy of 0.878 and a macro-F1 score of 0.875. The integrated verification service achieved an accuracy of 0.756 and a macro-F1 score of 0.752. On an end-to-end set of 234 samples, the system attained Recall@10 of 0.761, MRR of 0.612, and an accuracy of 0.578 on samples for which a substantive verdict was produced. GPU execution accelerated the verification module by a factor of 5.57 and the end-to-end server route by a factor of 3.07. The results demonstrate the value of decomposing automated fact-checking into interpretable stages and provide a quantitative basis for localizing error sources in a cascaded verification system.

**Keywords:** automated fact-checking, misinformation, natural language processing, evidence retrieval, natural language inference, RuBERT, vector search, explainable artificial intelligence.

## 1. Введение

Высокая скорость распространения информационных сообщений в интернет-СМИ и социальных сетях сформировала среду, в которой проверка фактов должна выполняться одновременно оперативно, масштабируемо и прозрачно для пользователя. Классическая бинарная классификация новостного документа по его лингвистическим признакам решает лишь часть задачи: она может фиксировать статистические закономерности корпуса, но не предъявляет доказательство, связывающее конкретное утверждение с проверяемым событием. Для прикладного фактчекинга принципиально важно не только определить метку, но и восстановить цепочку рассуждения: какое утверждение проверялось, какой источник был найден, какой фрагмент использован и какое логическое отношение установлено между утверждением и доказательством.

Современные исследования автоматизированной проверки фактов рассматривают задачу как последовательность взаимосвязанных стадий: извлечение проверяемых утверждений, поиск релевантных документов, отбор доказательных фрагментов и вынесение вердикта [2]. Доказательные архитектуры на основе внешнего поиска демонстрируют преимущество в динамичной информационной среде, поскольку позволяют актуализировать контекст без полного переобучения классификатора [6–9]. Вместе с тем качество полной системы не определяется одной моделью. Ошибка в ранней стадии изменяет вход всех последующих компонентов, а поэтому высокая точность семантического классификатора не гарантирует сопоставимого качества конечного документного вердикта.

В работе предлагается интеллектуальная система доказательной верификации русскоязычных новостных текстов. В отличие от подхода, при котором документ непосредственно классифицируется как достоверный или

недостоверный, исходный текст преобразуется в ограниченное множество атомарных утверждений. Для каждого утверждения формируется поисковый запрос, извлекаются публикации доверенных источников, выбираются релевантные фрагменты и определяется одно из трёх отношений логического следования: подтверждение, противоречие или нейтральность. Итоговый статус документа вычисляется на основе совокупности частных результатов с учётом силы доказательств и разнообразия источников.

**Цель исследования** состоит в разработке и многоуровневой экспериментальной оценке системы автоматизированной верификации, объединяющей извлечение фактов, поиск внешних доказательств и семантическое распознавание отношения «утверждение — доказательство».

Научная новизна исследования определяется следующими результатами. Предложен способ формирования русскоязычного корпуса логического следования на основе контролируемых искажений новостных публикаций; разработана каскадная процедура доказательной верификации с контролируемым отказом от категоричного вывода при отсутствии релевантных данных; предложена многоуровневая схема оценки, разделяющая качество NLI-модели, сервисной логики отбора доказательств и полного конвейера. Такая схема позволяет оценивать не только итоговую метрику, но и вклад каждой стадии в формирование результата.

Практическая значимость состоит в создании воспроизводимого программного прототипа, который возвращает пользователю общий статус текста, список выделенных утверждений, доказательные фрагменты, ссылки на источники и оценки уверенности. Модульная структура позволяет независимо развивать средства извлечения утверждений, поиска и семантической классификации, сохраняя единый интерфейс проверки.

## 2. Состояние исследований в области автоматизированного фактчекинга

Подходы к обнаружению недостоверной информации условно разделяются на три группы. Первая группа использует поверхностные и статистические признаки текста: частотные представления, стилистические характеристики, сведения об авторе и источнике. Такие методы эффективны при наличии устойчивого распределения данных, однако чувствительны к изменению тематики и способа формулировки сообщения. Российские исследования также подчёркивают необходимость проверять не только лексику, но и сущности, даты, числовые значения, цитаты и первоисточники [1, 5].

Вторая группа включает нейросетевые и трансформерные модели, обучаемые непосредственно по тексту документа. Перенос обучения и модели семейства BERT позволяют учитывать контекстные зависимости и адаптировать классификатор к специализированной предметной области [3, 4]. Однако документная метка по-прежнему остаётся результатом внутреннего вывода модели и не всегда сопровождается проверяемым обоснованием.

Третья группа объединяет доказательные методы, в которых вердикт строится с привлечением внешних документов. Системы AIC CTU и Evidence-backed Fact Checking показывают, что поиск доказательств и ограничение вывода содержанием найденных источников повышают обоснованность автоматической проверки [6, 7]. VeraCT Scan сочетает извлечение доказательств, оценку авторитетности источников и объяснимую классификацию [8]. Архитектура MUSER развивает многошаговый поиск, в котором запрос уточняется по мере накопления контекста [9]. Эти работы подтверждают продуктивность декомпозиции фактчекинга, но одновременно показывают, что модуль поиска становится самостоятельным источником ошибок.

Для семантического поиска широко применяются модели Sentence-BERT, преобразующие предложения и документы в плотные векторные представления

[10, 11]. Комбинация быстрого извлечения кандидатов и последующего переранжирования позволяет сохранить масштабируемость и повысить точность верхней части выдачи [12, 13]. Индекс HNSW обеспечивает эффективный приближённый поиск ближайших соседей в многомерном пространстве [14]. В предлагаемой системе эти идеи объединены с русскоязычной NLI-моделью и правилами агрегации, ориентированными на объяснимый документный вердикт.

## 3. Архитектура и алгоритмы доказательной верификации

### 3.1. Общая схема обработки

Система реализована как набор взаимодействующих сервисов, связанных серверным оркестратором. Функциональный конвейер включает четыре основные стадии: лингвистическую обработку исходного текста; поиск и индексирование доказательств; семантическую верификацию пар «утверждение — доказательство»; агрегацию частных результатов в документный вердикт. Потоки данных представлены на рис. 1.

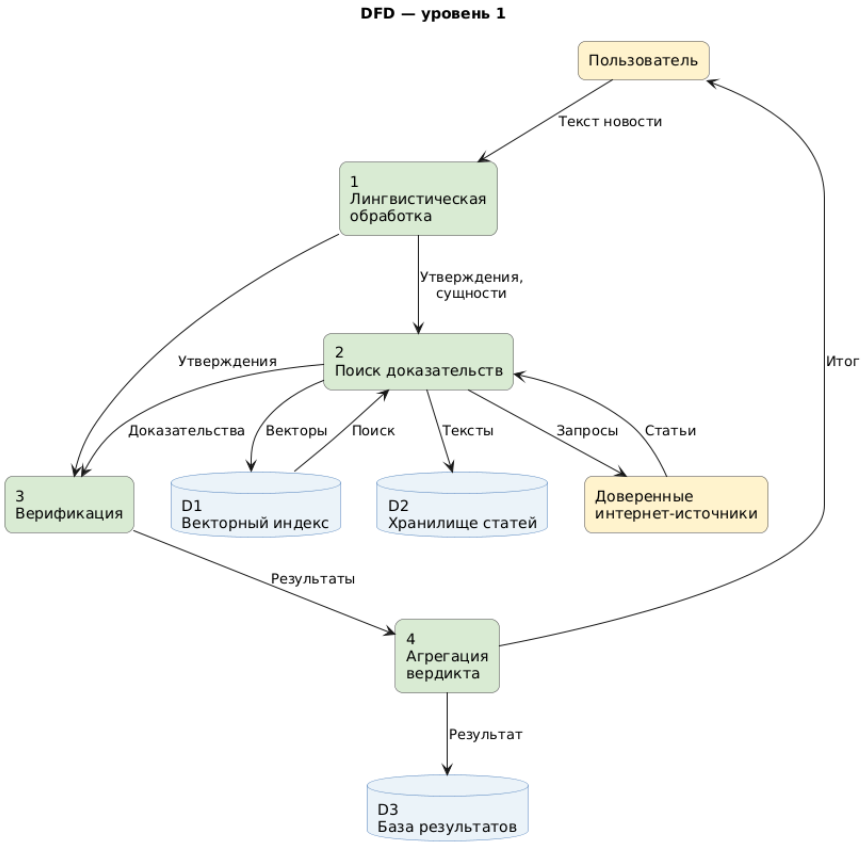

Рис. 1. Функциональная схема конвейера доказательной верификации

Поисковый модуль обращается к внешним источникам через метапоисковый агрегатор, извлекает основной текст публикаций и одновременно пополняет локальную доказательную базу. Полные тексты сохраняются в объектном хранилище, их фрагменты и векторные представления — в Qdrant [18], а структурированные результаты запросов — в реляционной базе данных. Такое разделение поддерживает повторное использование ранее обработанных материалов и обеспечивает аудит происхождения доказательства.

### 3.2. Выделение проверяемых утверждений

Лингвистический модуль преобразует новостной текст в набор атомарных высказываний, пригодных для сопоставления с внешними источниками. После очистки и нормализации выполняются сегментация на предложения, выделение именованных сущностей и разбиение сложных предложений на смысловые части. Вопросительные, условные, выраженно оценочные, слишком короткие и чрезмерно длинные фрагменты исключаются. Повторы удаляются по нормализованному тексту.

Для ранжирования применяется оценка фактологичности, учитывающая числа, даты, фактологические глаголы, количественные и финансовые показатели, именованные сущности и формальные признаки законченного высказывания. Оценка может быть представлена в виде

$$S_f(c) = 0{,}25I_{num} + 0{,}15I_{date} + 0{,}20I_{verb} + 0{,}15I_{quant} + \min(0{,}25;\ 0{,}08N_{ent}) + 0{,}05I_{cap},$$

где I_num, I_date, I_verb, I_quant и I_cap — индикаторы соответствующих признаков, N_ent — число найденных сущностей. Итоговое значение ограничивается единицей. Утверждения сортируются по оценке и количеству сущностей, после чего в поисковый модуль передаётся ограниченное множество наиболее информативных фрагментов. Ранжирование снижает

нагрузку на последующие стадии и концентрирует поиск на проверяемых фактах.

### 3.3. Поиск и индексирование доказательств

Для каждого утверждения формируются поисковые формулировки с использованием обнаруженных сущностей и контекста исходного сообщения. Результаты внешнего поиска через SearXNG [19] фильтруются по списку доверенных доменов. При недостаточности внешней выдачи используются RSS-ленты и локальный индекс ранее обработанных публикаций. Основной текст веб-страницы извлекается с удалением навигационных блоков, рекламы и повторяющихся элементов; инструментальная основа такого извлечения соответствует подходу библиотеки Trafilatura [15]. Почти идентичные публикации исключаются с использованием компактных хеш-представлений, развивающих подход SimHash [16].

Публикации разбиваются на перекрывающиеся фрагменты. Для каждого фрагмента вычисляется 384-мерное векторное представление, которое сохраняется вместе с URL, доменом, заголовком, датой публикации и коэффициентом доверия к источнику. Поиск выполняется как извлечение ближайших векторов с последующей фильтрацией и переранжированием. В отличие от одноразового веб-поиска, эта схема формирует накапливаемую локальную базу доказательств и сохраняет происхождение каждого фрагмента.

### 3.4. Семантическая верификация и агрегация вердикта

Центральным компонентом является NLI-классификатор, который получает пару «доказательный фрагмент — утверждение» и возвращает вероятности классов ENTAILS, CONTRADICTS и NEUTRAL. В качестве основы использована модель cointegrated/rubert-base-cased-nli-threeway [17]. Пары обрабатываются пакетно, что снижает накладные расходы токенизации и позволяет эффективно использовать GPU.

Перед классификацией фрагменты повторно ранжируются относительно конкретного утверждения. Если максимальная релевантность не достигает

установленного порога, система фиксирует отсутствие достаточного доказательства и не формирует категоричную метку. Противоречие выбирается только при одновременном выполнении двух условий: вероятность класса превышает минимальную уверенность и имеет заданный отрыв от вероятности подтверждения. Такая логика реализует контролируемый отказ от ответа и снижает риск ложного обвинения публикации в недостоверности.

Документный статус вычисляется по совокупности утверждений. Для статуса REFUTED учитываются доля и минимальное число противоречащих утверждений, а также количество различных доменов, предоставивших доказательства. Статус CONFIRMED формируется при достаточном преобладании подтверждающих результатов. Если релевантные источники не найдены или оценки модели не позволяют сделать устойчивый вывод, возвращается INSUFFICIENT_DATA. Пользователь получает не только итог, но и детализацию по каждому утверждению, что обеспечивает интерпретируемость решения.

## 4. Формирование корпуса и методика эксперимента

### 4.1. Специализированный корпус логического следования

Для адаптации RuBERT сформирован корпус пар «утверждение — доказательство». Он объединяет синтетические примеры, построенные по доверенным новостным публикациям, и примеры ручной разметки. Синтетическая часть создавалась локальной моделью Qwen 3 8B. Модель получала заголовок и текст публикации и формировала учебную новость с одним контролируемым фактическим искажением. Одновременно сохранялись противоречащее предложение и дословная цитата из исходной статьи, позволяющая его опровергнуть.

Использовались пять типов искажений: прямое противоречие, изменение числового значения, замена сущности, изменение причинно-следственной связи и инверсия цитаты. Ограничение генерации содержанием исходной публикации предотвращало добавление внешних фактов. Результат формировался в

структурированном формате, содержащем синтетический текст, проверяемое утверждение, доказательную цитату и тип искажения.

Автоматическая фильтрация проверяла наличие обязательных полей, допустимость типа искажения, вхождение утверждения в синтетический текст и присутствие доказательства в исходной статье. Для класса CONTRADICTS использовались искажённое утверждение и опровергающая цитата; для ENTAILS — согласованные фрагменты исходной публикации; нейтральные и дополнительные пары дополнялись ручной разметкой. После удаления дубликатов корпус включал 391 пример. Стратифицированное распределение по подвыборкам представлено в табл. 1.

Таблица 1. Состав корпуса для обучения RuBERT

| **Раздел** | **Размер** | **CONTRADICTS** | **ENTAILS** | **NEUTRAL** |
|---|---|---|---|---|
| Общий корпус | 391 | 133 | 133 | 125 |
| Обучающая часть | 312 | 106 | 106 | 100 |
| Валидационная часть | 38 | 13 | 13 | 12 |
| Тестовая часть | 41 | 14 | 14 | 13 |

### 4.2. Настройка модели и уровни оценки

Дообучение выполнялось в течение четырёх эпох со скоростью обучения $5 \cdot 10^{-6}$, коэффициентом регуляризации весов 0,01 и долей линейного прогрева 0,05. Максимальная длина входной последовательности составляла 256 токенов. Размер обучающего пакета был равен одному примеру с накоплением градиента до восьми примеров. Невысокая скорость обучения выбрана для

сохранения знаний исходной NLI-модели при адаптации к специализированной новостной области.

Экспериментальная методика разделяет три уровня. На первом уровне оценивается собственно RuBERT на заранее подготовленных парах. На втором уровне проверяется модуль верификации, который дополнительно выполняет отбор фрагментов и применяет пороговую логику. На третьем уровне анализируется полный путь от исходного текста до документного статуса. Такое разделение превращает итоговую оценку в диагностический инструмент: разница между уровнями количественно характеризует влияние поиска, подготовки доказательств и агрегации.

Для NLI использовались Accuracy, Macro-F1 и F1 по классам. Качество поиска оценивалось по Recall@1, Recall@10 и MRR. Для сквозного конвейера дополнительно рассчитывались доля образцов с содержательным вердиктом (coverage) и Accuracy на покрытых образцах. Последняя метрика отражает качество решения при условии, что система располагает достаточной доказательной базой.

### 4.3. Экспериментальное окружение

Сервисы развёртывались в контейнерном окружении. Производительность сравнивалась в двух режимах: принудительное выполнение на CPU и использование CUDA-версии PyTorch с доступом к GPU. Внешний сетевой поиск отключался, а доказательства извлекались из одного и того же локального Qdrant-индекса, благодаря чему измерение отражало именно вычислительную составляющую. Для каждого этапа фиксировались средняя задержка и 95-й перцентиль. Контейнеризация обеспечивала согласованность версий библиотек и параметров исполнения в обоих режимах [20].

## 5. Результаты

### 5.1. Качество распознавания логического следования

Результаты тестирования RuBERT и интегрированного модуля приведены в табл. 2. Дообученная модель достигла Macro-F1 0,875. Наиболее высокий показатель получен для класса NEUTRAL, а F1 класса CONTRADICTS составила 0,800. Это особенно важно для фактчекинга, поскольку именно корректное распознавание противоречия определяет способность системы опровергать искажённое утверждение.

Таблица 2. Качество модели и модуля верификации

| **Показатель** | **RuBERT** | **Модуль верификации** |
|---|---|---|
| Число примеров | 41 | 41 |
| Accuracy | 0,878 | 0,756 |
| Macro-F1 | 0,875 | 0,752 |
| F1 ENTAILS | 0,897 | 0,667 |
| F1 CONTRADICTS | 0,800 | 0,800 |
| F1 NEUTRAL | 0,929 | 0,788 |
| Средняя скорость / задержка | 54,2 прим./с | 228,4 мс |

Сохранение F1 класса CONTRADICTS на уровне 0,800 после включения сервисной логики показывает устойчивость наиболее значимого для опровержения класса. Изменение общих метрик связано прежде всего с тем, что модуль верификации оценивает не подготовленную пару напрямую, а сначала выбирает фрагмент и применяет пороги релевантности и уверенности. Следовательно, разность между двумя столбцами отражает работу механизма контролируемого отбора доказательств, а не только классификатора.

### 5.2. Сквозная оценка системы

Полный конвейер обработал 234 образца без технических ошибок. Результаты представлены в табл. 3. Recall@10, равный 0,761, показывает, что релевантное доказательство в большинстве случаев присутствует среди первых десяти результатов. Значение MRR 0,612 характеризует достаточно высокое положение первого релевантного фрагмента в выдаче.

Таблица 3. Сквозная оценка на выборке REFUTED/CONFIRMED

| Показатель | Значение |
|---|---|
| Число образцов | 234 |
| Ошибки выполнения | 0 |
| End-to-end Accuracy | 0,397 |
| End-to-end Macro-F1 | 0,237 |
| Coverage без INSUFFICIENT_DATA | 68,8 % |
| Accuracy на покрытых образцах | 0,578 |
| Retrieval Recall@1 | 0,538 |
| Retrieval Recall@10 | 0,761 |
| MRR поиска | 0,612 |

Содержательный вердикт сформирован для 68,8 % образцов; на этой части Accuracy составила 0,578. Остальные случаи были корректно переведены в состояние INSUFFICIENT_DATA, когда система не располагала достаточным сочетанием релевантности и уверенности. Такой результат демонстрирует практическое значение трёхстатусной схемы: вместо принудительного бинарного решения конвейер явно отделяет отсутствие доказательств от подтверждения или опровержения.

### 5.3. Вычислительная эффективность

Использование GPU дало наибольший эффект на стадиях, содержащих плотные матричные вычисления. Результаты сравнения приведены в табл. 4. Модуль верификации ускорился в 5,57 раза, а средняя задержка полного маршрута /verify сократилась на 67,4 %. Поисковый модуль получил ускорение

1,35 раза. Лингвистическая обработка, основанная преимущественно на правилах и сегментации, ожидаемо не выиграла от переноса на GPU.

Таблица 4. Производительность CPU- и GPU-сборок

| Этап | CPU, мс | GPU, мс | Кратность | Снижение, % | CPU P95 | GPU P95 |
|---|---|---|---|---|---|---|
| Backend /verify | 4 322,38 | 1 408,66 | 3,07 | 67,4 | 10 258,75 | 21 89,85 |
| NLP-service | 1 66,67 | 1 86,39 | 0,89 | −11,8 | 28 9,92 | 34 4,68 |
| Search-service | 1 29,38 | 9 5,64 | 1,35 | 26,1 | 28 8,60 | 17 1,42 |
| Verifier-service | 8 67,15 | 1 55,78 | 5,57 | 82,0 | 14 39,80 | 23 9,43 |

## 6. Обсуждение результатов

Многоуровневая оценка показывает, что разработанная система решает две различные, но связанные задачи. На уровне подготовленных пар она точно распознаёт логическое отношение между утверждением и доказательством. На уровне полного конвейера она должна дополнительно найти проверяемое утверждение, сформировать запрос, получить документ, выбрать фрагмент и агрегировать несколько частных решений. Поэтому разрыв между Macro-F1 NLI-модели 0,875 и сквозной метрикой является количественной характеристикой каскада и позволяет локализовать направления дальнейшего развития.

Первый значимый результат связан с поиском. Recall@1 0,538 и Recall@10 0,761 означают, что расширение окна кандидатов существенно увеличивает вероятность присутствия нужного доказательства. Это обосновывает двухэтапную схему «извлечение — переранжирование»: быстрый векторный индекс формирует множество кандидатов, после чего

более точная модель выбирает фрагменты для NLI. Значение MRR 0,612 подтверждает, что релевантные материалы обычно расположены в верхней части выдачи.

Второй результат относится к контролируемому отказу от ответа. Состояние INSUFFICIENT_DATA не является нейтральной разновидностью бинарного класса: оно сообщает о качестве доказательной базы. Coverage 68,8 % совместно с Accuracy 0,578 на покрытых образцах позволяет отдельно оценивать способность системы найти основание для решения и корректность решения при наличии такого основания. Для прикладного фактчекинга эта декомпозиция важнее принудительной классификации каждого сообщения.

Третий результат состоит в устойчивом распознавании противоречий. F1 класса CONTRADICTS сохраняется на уровне 0,800 как для модели, так и для модуля верификации. Этому способствует специализированный корпус с контролируемыми типами искажений. Изменение числа, сущности, причины или направления цитаты формирует обучающие примеры, близкие к реальным способам смысловой модификации новостей и не сводимые к поверхностным стилистическим признакам.

Архитектурная декомпозиция имеет и эксплуатационное значение. Ускорение verifier-service в 5,57 раза показывает, что ресурсоёмкая NLI-классификация может масштабироваться независимо от остальных компонентов. Одновременно система сохраняет трассируемость: документный статус связан с конкретными утверждениями, доказательствами, URL и вероятностями классов. Таким образом, программная реализация объединяет вычислительную эффективность и объяснимость результата.

Предложенная методика применима не только к рассматриваемой реализации. Разделение оценки на модельный, сервисный и сквозной уровни может использоваться для сравнения поисковых стратегий, моделей эмбеддингов, переранжировщиков и правил агрегации. Сохранение одинаковой структуры метрик позволяет определять, на какой стадии возникает изменение

качества, и выбирать улучшения на основании измерений, а не только итоговой точности.

## 7. Заключение

Разработана интеллектуальная система доказательной верификации русскоязычных новостных текстов, объединяющая извлечение атомарных утверждений, поиск по доверенному информационному пространству, векторный отбор доказательств, NLI-классификацию и объяснимую агрегацию вердикта. В отличие от изолированной классификации документа система сохраняет полную цепочку проверки и предоставляет пользователю основания каждого частного решения.

Сформирован специализированный корпус из 391 пары по трём классам логического следования. Контролируемое синтетическое построение позволило представить пять типов фактологических искажений, а ручная разметка обеспечила баланс подтверждающих, противоречащих и нейтральных примеров. Дообученная RuBERT достигла Macro-F1 0,875; модуль верификации — 0,752. Сквозной эксперимент подтвердил работоспособность полного конвейера на 234 образцах, продемонстрировал Recall@10 0,761 и Accuracy 0,578 на покрытых образцах.

Предложенная многоуровневая методика превратила экспериментальную оценку в средство диагностики каскадной архитектуры. Она показала, что качество конечного вердикта формируется совместным действием лингвистической обработки, поиска, отбора фрагментов, семантической классификации и пороговой агрегации. GPU-оптимизация дополнительно обеспечила ускорение центрального модуля в 5,57 раза. Полученные результаты создают основу для развития русскоязычных доказательных систем фактчекинга, расширения доказательной базы и интеграции решения в редакционные и аналитические процессы.

## Список литературы

1. Комлева В. Ю., Соломин В. Е. Алгоритм фактчекинга в работе региональных сетевых СМИ // Виртуальная коммуникация и социальные сети. 2022. Т. 1, № 4. С. 167–171.

2. Guo Z., Schlichtkrull M., Vlachos A. A survey on automated fact-checking // Transactions of the Association for Computational Linguistics. 2022. Vol. 10. P. 178–206.

3. Baashirah R. Zero-shot automated detection of fake news: an innovative approach (ZS-FND) // IEEE Access. 2024. Vol. 12. P. 182828–182840.

4. Fazlija B., Bakiji I., Dauti V. FinFakeBERT: financial fake news detection // Frontiers in Artificial Intelligence. 2025. Vol. 8. Art. 1604272.

5. Мукаев И. Р., Кочетков И. В. Метод выявления недостоверной информации на основе синтактико-семантического анализа с учетом контекста сообщения // НеоКВЕСТ. 2025. № 2. С. 35–37. EDN IVDQMS.

6. Ullrich H., Mlynář T., Drchal J. AIC CTU system at AVeriTeC: Re-framing automated fact-checking as a simple RAG task // Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER). 2024. P. 137–150.

7. Singal R. et al. Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs // Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER). 2024. P. 91–98.

8. Niu C. et al. VeraCT Scan: Retrieval-augmented fake news detection with justifiable reasoning // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics. Vol. 3: System Demonstrations. 2024. P. 266–277.

9. Liao H. et al. MUSER: A multi-step evidence retrieval enhancement framework for fake news detection // Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023. P. 4461–4472.

10. Reimers N., Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks // Proceedings of EMNLP-IJCNLP. 2019. P. 3982–3992.

11. Reimers N., Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation // Proceedings of EMNLP. 2020. P. 4512–4525.

12. Gao L., Dai Z., Callan J. Rethink training of BERT rerankers in multi-stage retrieval pipeline // Advances in Information Retrieval. ECIR 2021. P. 280–286.

13. Nair S. et al. Transfer learning approaches for building cross-language dense retrieval models // Advances in Information Retrieval. ECIR 2022. P. 382–396.

14. Malkov Y. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020. Vol. 42, no. 4. P. 824–836.

15. Barbaresi A. Trafilatura: A web scraping library and command-line tool for text discovery and extraction // Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics: System Demonstrations. 2021. P. 122–131.

16. Charikar M. S. Similarity estimation techniques from rounding algorithms // Proceedings of the 34th Annual ACM Symposium on Theory of Computing. 2002. P. 380–388.

17. Cointegrated. RuBERT for NLI: cointegrated/rubert-base-cased-nli-threeway [Электронный ресурс]. URL: https://huggingface.co/cointegrated/rubert-base-cased-nli-threeway (дата обращения: 10.04.2026).

18. Qdrant Documentation [Электронный ресурс]. URL: https://qdrant.tech/documentation/ (дата обращения: 10.04.2026).

19. SearXNG Documentation [Электронный ресурс]. URL: https://docs.searxng.org/ (дата обращения: 10.04.2026).

20. Merkel D. Docker: lightweight Linux containers for consistent development and deployment // Linux Journal. 2014. Vol. 2014, no. 239. Art. 2.