

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ/ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

DOI: https://doi.org/10.60797/IRJ.2026.168.70 EDN: ZUEYPW

ОПТИМИЗАЦИЯ КОНТЕКСТА БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ В АГЕНТНЫХ СИСТЕМАХ: ОТ СТАТИЧЕСКОГО ПРОМПТ-ИНЖИНИРИНГА К ДИНАМИЧЕСКОЙ ИНЖЕНЕРИИ КОНТЕКСТА

Научная статья

Лабинцев А.И.[1, *], Мыратгелдиев А.[2]
[1] ORCID : 0000-0002-5167-2689;
[2] ORCID : 0009-0008-3688-5679;
[1, 2] Финансовый университет при Правительстве Российской Федерации, Москва, Российская Федерация

* Корреспондирующий автор (andrej.labintsev[at]yandex.ru)

Предложена: 21.04.2026; Принята: 22.05.2026; Опубликована: 17.06.2026

Аннотация

В статье предложена формальная математическая модель оптимизации контекста в агентных системах — как многокритериальной задачи максимизации ожидаемого вознаграждения при ограничениях на длину контекста, объём данных, вычислительные ресурсы и качество источников. Новизна исследования состоит в возможности применять методы математической оптимизации (например, методы нелинейного программирования, эволюционные алгоритмы или reinforcement learning) для автоматизированного подбора оптимальной стратегии обогащения контекста — с учётом баланса качества ответа и ресурсных затрат.

В ходе экспериментального исследования проведена апробация предложенной математической модели. Выполнена оптимизация контекста на пяти сценариях работы LLM (простой промпт, RAG на неструктурированных данных, поиск по структурированным данным, вызов tools, механизм памяти диалога) на двух моделях (Qwen3-Ru и Qwen3.5) на примере задачи разработки консультанта для поступающих в ВУЗ. Оценка качества выполнялась методом LLM as Judge по шкале 0–9. Установлено, что наибольший прирост качества обеспечивают вызов инструментов (Δ до +3,5 балла относительно простого промпта) и механизм памяти диалога (Δ до +3,7 балла). Показано, что архитектурные решения влияют на качество сильнее, чем простое увеличение объёма контекста. Qwen3.5 демонстрирует лучшее соотношение качества и вычислительной эффективности (время выполнения тестового набора — 2 минуты против 9 минут у Qwen3-Ru).

Результаты подтверждают, что ключевым фактором эффективности LLM в прикладных задачах является не масштаб контекста, а способ его организации, структурирования и интеграции с внешними инструментами и памятью.

Ключевые слова: большие языковые модели, агентные системы, инженерия контекста, оптимизация контекста, промпт-инжиниринг, информационный домен, RAG, вызов инструментов, память диалога, структурированные данные, LLM as Judge, многокритериальная оптимизация, качество генерации, вычислительная эффективность, Qwen3-Ru, Qwen3.5.

OPTIMISING THE CONTEXT OF LARGE LANGUAGE MODELS IN AGENT-BASED SYSTEMS: FROM STATIC PROMPT ENGINEERING TO DYNAMIC CONTEXT ENGINEERING

Research article

Labincev A.I.[1, *], Myratgeldiyev A.[2]
[1] ORCID : 0000-0002-5167-2689;
[2] ORCID : 0009-0008-3688-5679;
[1, 2] Financial University under the Government of the Russian Federation, Moscow, Russian Federation

* Corresponding author (andrej.labintsev[at]yandex.ru)

Suggested: 21.04.2026; Accepted: 22.05.2026; Published: 17.06.2026

Abstract

The article proposes a formal mathematical model for context optimisation in agent-based systems as a multi-criteria problem of maximising expected reward with constraints on context length, data volume, computational resources and source quality. The novelty of the research consists in the possibility of applying mathematical optimisation methods (e.g. non-linear programming methods, evolutionary algorithms or reinforcement learning) for the automated selection of an optimal context enrichment strategy taking into account the balance between response quality and resource costs.

The suggested mathematical model was tested as part of an experimental study. Context optimisation was performed across five LLM operation scenarios (simple prompt, RAG on unstructured data, search on structured data, tool calling, dialogue memory mechanism) on two models (Qwen3-Ru and Qwen3.5), based on the example of developing a consultant for university applicants. Quality assessment was performed using the LLM as Judge method on a scale of 0–9. It was found that the greatest improvement in quality was achieved by tool invocation (Δ up to +3.5 points relative to a simple prompt) and the dialogue memory mechanism (Δ up to +3.7 points). It is shown that architectural solutions have a greater impact on quality than simply increasing the context size. Qwen3.5 demonstrates a better balance between quality and computational efficiency (test set execution time of 2 minutes compared to 9 minutes for Qwen3-Ru).

The results confirm that the key factor in the effectiveness of LLMs for practical tasks is not the scale of the context, but rather the way it is organised, structured and integrated with external tools and memory.

Keywords: large language models, agent-based systems, context engineering, context optimisation, prompt engineering, information domain, RAG, tool calling, dialogue memory, structured data, LLM as Judge, multi-criteria optimisation, generation quality, computational efficiency, Qwen3-Ru, Qwen3.5.

Введение

Появление больших языковых моделей (Large Language Models, LLMs) продемонстрировало беспрецедентные возможности в понимании естественного языка, его генерации и рассуждении. LLM эволюционировали от базовых систем, выполняющих инструкции, до центральных механизмов рассуждений в сложных агентных системах.

Однако производительность и эффективность этих моделей определяется не только информацией, которую они получают на этапе обучения, но и контекстом, предоставляемым на этапе инференции (вывода). По мере усложнения решаемых задач развивались методы проектирования и управления информацией — как на этапе обучения, так и на этапе предсказания.

Информационный домен — это область знаний или сфера деятельности, в рамках которой функционирует LLM и для которой требуется специфический набор данных, терминологии, правил и закономерностей. Информационный домен задаёт границы и специфику контекста, в котором модель должна демонстрировать компетентность. Например:

- медицина (термины, протоколы лечения, научные исследования);
- юриспруденция (законы, прецеденты, процессуальные нормы);
- финансы (рыночные показатели, экономические теории, нормативные акты);
- техническое образование (инженерные дисциплины, стандарты, методики обучения);
- кибербезопасность (угрозы, протоколы защиты, анализ кода).

В рамках каждого информационного домена требуются специфические подходы к формированию входных данных для LLM, поскольку универсальные методы могут не учитывать нюансы терминологии, логики рассуждений и структуры знаний конкретной области.

Статические инструкции для LLM принято называть промптом (prompt), а методы их формирования изучают в дисциплине промпт-инжиниринга (Prompt Engineering). Эта область фокусируется на разработке оптимальных формулировок запросов, позволяющих добиться от модели точных и релевантных ответов в рамках заданного информационного домена.

Динамические методы формирования подсказок, учитывающие текущий контекст, внешние источники знаний и историю взаимодействия, изучаются в рамках инженерии контекста (Context Engineering). В отличие от статических промптов, контекстные подсказки могут включать фрагменты релевантных документов из внешних баз знаний, исторические диалоги или предыдущие шаги рассуждения, структурированные данные (таблицы, графы знаний) и т.д.

Применение методов инженерии контекста позволяет дополнить знания агента актуальной информацией за пределами предобученной базы, направить его поведение в нужное русло с учётом специфики информационного домена, повысить точность и релевантность ответов за счёт интеграции внешних данных и снизить вероятность «галлюцинаций» (вымышленных фактов) в генерации.

Цель работы — раскрыть возможности агентных систем на базе LLM в полной мере, повысить качество диалогов и решений в заданном информационном домене за счёт применения методов контекстной инженерии. В своей работы мы разработали методику оптимизации контекста LLM к специфике предметной области и протестировали эффективность на примере создания консультанта для поступающих в высшее учебное заведение.

Исследования последних лет демонстрируют растущий интерес к способам повышения эффективности больших языковых моделей (LLM) за счёт оптимизации контекста, подаваемого на вход.

Ранние работы [1] в этой области были сосредоточены преимущественно на дизайне промптов (prompt engineering) — подборе формулировок запросов, которые позволяют добиться от модели более точных и релевантных ответов. В рамках этого направления были разработаны такие техники, как few-shot prompting, chain-of-thought (CoT) [2] и zero-shot CoT, показавшие, что структурирование запроса может существенно улучшить качество генерации. Эти методы легли в основу современных подходов к контекстной инженерии [3].

Параллельно развивались подходы, предполагающие интеграцию LLM с внешними источниками знаний. Ключевым прорывом здесь стала концепция Retrieval-Augmented Generation (RAG) [4], объединяющая возможности поиска релевантной информации (retrieval) и генерации текста на основе найденных данных. Исследования в этой сфере заложили основы для создания более сложных архитектур, включая модульные системы и агентные архитектуры, где LLM взаимодействует с инструментами поиска и базами данных.

В русскоязычной научной литературе также активно исследуются различные аспекты RAG и контекстной инженерии:

Оболенский Д. М. [5] рассматривает применение RAG в интеллектуальных образовательных экосистемах с использованием библиотеки LangChain, языковой модели GigaChat и векторной СУБД Qdrant. Система обрабатывает описания вакансий и образовательных ресурсов для генерации персонализированных описаний рынка труда. Проведен анализ публикационной активности и научных коллабораций научно-педагогических работников [6]. Оценка использования GigaCode в деятельности IT-компаний [7] включает сравнение с аналогичными решениями, такими как GitHub Copilot и Amazon CodeWhisperer.

Науменко А. О. [8] анализирует архитектуру RAG, её преимущества и ограничения по сравнению с традиционными методами обучения LLM. Автор описывает ключевые компоненты RAG (методы индексации, поиска и интеграции информации) и подчёркивает значимость технологии для повышения точности и надёжности генерируемого контента.

Волков С. С., Шалыгин С. В., Лабинцев А. И. [9] исследуют оптимизацию контекста LLM в высшем техническом образовании, предлагая подходы к адаптации моделей для образовательных задач.

Значительный объём работ посвящён решению проблемы обработки длинных последовательностей — одной из ключевых сложностей при работе с расширенным контекстом. Предложены методы сжатия контекста, иерархического управления памятью и селективного извлечения информации, позволяющие моделям эффективно оперировать большими объёмами данных без потери производительности:

Гисин В. Б. [10] предлагает динамическую модель внимания в трансформерах, улучшающую обработку длинных последовательностей.

Болтачев Э. Ф., Фархадов М. П., Тюляков А. И. [11] исследуют методы токенизации текстов в финансовой сфере, что напрямую влияет на эффективность представления контекста для LLM.

Особое внимание уделяется вопросам безопасности и надёжности LLM:

Унижаев Н. В. [12] анализирует модель угроз конфиденциальной информации в LLM, выявляя риски утечки данных при работе с контекстом.

Швыров В. В., Капустин Д. А., Сентяй Р. Н. [13] предлагают методы использования LLM с поддержкой рассуждений для анализа безопасности программного кода, демонстрируя возможности контекстной инженерии в прикладных задачах.

Несмотря на существенный прогресс, анализ более 1400 исследований [3] выявил критическую асимметрию в возможностях современных LLM. Модели, усиленные методами контекстной инженерии, демонстрируют впечатляющую способность понимать сложные и объёмные контексты. Однако, при генерации развёрнутых, логически связных и детализированных текстов они сталкиваются с заметными ограничениями: ответы могут терять последовательность, содержать фактические ошибки или излишне повторяться. Этот разрыв между способностями к восприятию и генерации представляет собой важнейший нерешённый вопрос, определяющий направления будущих исследований в области контекстно-ориентированного ИИ.

Методы и принципы исследования

Пусть имеется некоторый набор задач, которые необходимо решить с помощью агента. Например: написать код на языке Python, проконсультировать покупателя или поступающего в ВУЗ и т.д. Множество задач в таком наборе теоретически бесконечно, однако на практике мы имеем дело с ограниченным набором двоек «запрос — ответ»:

$$T = \{(\tau, Y_\tau^*)\}$$

где:

τ — конкретный экземпляр задачи (запрос пользователя);

Y_τ^* — условно правильный ответ на запрос.

Вероятностная авторегрессионная (большая языковая) модель генерирует выходную последовательность путём максимизации условной вероятности:

$$P_\theta(Y|C) = \prod_{t=1}^{T} P_\theta(y_t|y_{<t}, C(\tau)) \quad (1)$$

где:

$C(\tau)$ — контекст, который используется для решения задачи;

θ — параметры языковой модели.

В инженерии промптов контекст C формируется как композиция запроса пользователя и статичной инструкции по решению задачи. В инженерии контекста C представляет собой динамически структурированный набор информационных компонентов $c_1, c_2, \dots, c_n$. Эти компоненты извлекаются из различных источников, фильтруются (отбираются по релевантности) и форматируются (приводятся к нужному виду) с помощью множества функций F.

$$\begin{cases} c_1 = f_1(\tau, D) \\ c_2 = f_2(\tau, D) \\ \dots \\ c_n = f_n(\tau, D) \end{cases} \quad (2)$$

где:

D — набор данных, необходимый для решения набора задач T;

c_i — различные компоненты информации. Например:

c_{prompt} — системные инструкции и правила для извлечения и генерации контекста;

$c_{retrieve}$ — внешние знания, извлекаемые с помощью механизмов RAG или графов знаний;

c_{tools} — описания и сигнатуры доступных внешних инструментов (вызов функций и рассуждения с интеграцией инструментов);

c_{reason} — информация из промежуточных рассуждений модели;

c_{memory} — сохраняемая информация из предыдущих взаимодействий (системы памяти и управления контекстом);

c_{state} — динамическое состояние диалога с пользователем, внешнего мира или многоагентной системы.

Конечный результат формируется за несколько итераций извлечения информации и генерации промежуточных рассуждений. Максимизация ожидаемого качества вывода агента формализуется как задача оптимизации. Пространство поиска включает в себя множество функций генерации и композиции контекста F.

Тогда целевая функция имеет вид:

$$F^* = \arg\max_F \mathbb{E}_{\tau \sim T} \left[\mathrm{Reward}(P_\theta(Y|C(F(\tau, D))), Y_\tau^*) \right] \quad (3)$$

где:

F^* — оптимальный набор функций генерации контекста;

τ — конкретный экземпляр задачи;

$C_F(\tau)$ — контекст, сгенерированный функциями из набора F для данной задачи τ ;

Y_τ^* — эталонный (идеальный) результат для задачи (ground-truth output);

$P_\theta(Y|C_F(\tau))$ — распределение вероятностей выходных последовательностей Y, генерируемых моделью с параметрами при заданном контексте;

$Reward$ — функция вознаграждения (метрика качества), оценивающая, насколько сгенерированный моделью вывод соответствует эталонному результату;

$\mathbb{E}_{\tau \sim T}$ — математическое ожидание по распределению задач, то есть усреднение по всем возможным задачам из рассматриваемого набора;

$\arg\max_F$ — операция поиска такого набора функций F, который максимизирует среднее вознаграждение.

Эта задача оптимизации имеет ряд ограничений.

1) Ограничение на длину контекста модели:

$$|C| \leq L_{\max}$$

где:

$|C|$ — длина (объём) контекста C (обычно измеряется в токенах);

$L_{\max}$ — максимально допустимая длина контекста для конкретной модели (например, 4 096, 8 192 или 32 000 токенов).

Это ограничение частично компенсируется сжатием информации (summarization), селективным отбором наиболее релевантных фрагментов и методами управления иерархической памятью.

2) Ограничение на объём доступных документов.

$$D' \subseteq D, \quad |D'| \leq V_{\max}$$

где:

D' — подмножество доступных документов;

$V_{\max}$ — максимальный объём данных, доступных для обработки (например, из-за ограничений хранилища, скорости доступа или стоимости API-запросов).

3) Ограничение на выборку задач.

$$T' \subset T, \quad |T'| \ll |T|$$

где:

T' — выборка задач, используемая для оценки качества системы;

$|T'|$ — размер выборки;

$|T|$ — полный объём возможного множества задач.

Ограниченная выборка не в полной мере отражает реальное распределение задач и недостаточно покрывает краевые случаи (edge cases).

4) Вычислительные ограничения.

$$\mathrm{Time}(F) \leq T_{\max}, \quad \mathrm{Cost}(F) \leq C_{\max}$$

где:

$\mathrm{Time}(F)$ — время выполнения функций генерации контекста;

$T_{\max}$ — максимально допустимое время ответа;

$\mathrm{Cost}(F)$ — стоимость вычислений (включая API-запросы, вычислительные ресурсы);

$C_{\max}$ — бюджет на обработку одной задачи.

Эти ограничения в совокупности формируют многокритериальную оптимизационную задачу, где необходимо балансировать между:

- качеством ответа;
- объёмом используемого контекста;
- затрат на сбор данных;
- репрезентативностью выборки;
- вычислительными ресурсами.

Таким образом, исследование направлено на оценку эффективности различных подходов к управлению контекстом в больших языковых моделях при решении предметно-ориентированных задач. В рамках данной работы для решения задачи оптимизации применяется метод полного перебора.

Основные результаты

В качестве тестовых моделей выбраны две LLM:

- Qwen3 с адаптацией к русскому языку [14];
- Qwen3.5 без адаптации, но с улучшенной архитектурой [15].

Для каждой модели тестируются пять сценариев взаимодействия, чтобы понять, как разные методы работы с контекстом влияют на качество ответов:

1. Простой промпт — модель получает только запрос пользователя и базовую инструкцию без дополнительного контекста.

2. RAG на неструктурированных данных — к запросу добавляется контекст из внешних источников, разбитый на чанки фиксированного размера.

3. Поиск по структурированным данным — модель использует заранее подготовленные структурированные данные (например, таблицы, JSON) для формирования ответа.

4. Вызов tools — модель может вызывать внешние инструменты (функции) для получения актуальной информации (например, поиск данных в таблицах).

5. Механизм памяти диалога — модель учитывает историю взаимодействия с пользователем, чтобы давать более согласованные и последовательные ответы.

Оценка качества ответов выполняется методом LLM as Judge [16]. Для этого используется отдельная языковая модель в роли эксперта. Ей подаются на вход эталонный ответ (ground-truth) и ответ тестируемой модели (response).

Шкала оценок:

0 — ответ полностью не соответствует эталонному (неверная информация, нерелевантен);

1–3 — существенные ошибки или пропуски, основная суть частично угадана;

4–6 — в целом релевантный ответ, но есть неточности, неполнота или небольшие ошибки;

7–8 — хороший ответ, близкий к эталонному, незначительные недочёты;

9 — практически идентичен эталонному, без ошибок.

Каждый сценарий тестируется на выборке из 30 типовых задач (например, консультации по поступлению в вуз, поиск стоимости обучения, уточнение количества мест).

Для каждого ответа вычисляется оценка по указанной шкале, затем рассчитывается средний балл по сценарию и модели.

Таблица 1 - Средние оценки качества ответов

DOI: https://doi.org/10.60797/IRJ.2026.168.70.1

Сценарий	Qwen3-Ru	Qwen3.5
Простой промпт	4,0	3,8
RAG на неструктурированных данных	5,1	4,8
Поиск по структурированным данным	5,1	5,1
Вызов tools (поиск по таблицам)	7,1	7,3
Механизм памяти диалога	7,3	7,5

Примечание: по шкале 0–9

Рисунок 1 - Сравнение качества ответов при различных сценариях управления контекстом

DOI: https://doi.org/10.60797/IRJ.2026.168.70.2

Среднее время выполнения тест кейса из 30 задач: Qwen3-Ru - 9 минут, Qwen3.5 - 2 минуты.

Таблица 2 - Распределение оценок по диапазонам

DOI: https://doi.org/10.60797/IRJ.2026.168.70.3

Диапазон оценок	Qwen3-Ru	Qwen3.5
0-3	3	0
4-6	2	5
7-8	16	21
9	7	2

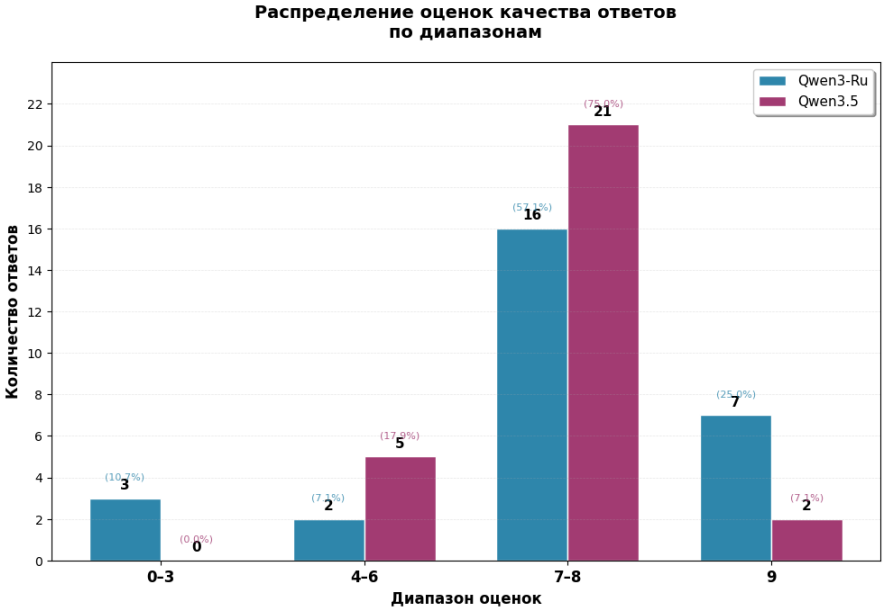

Рисунок 2 - Распределение оценок качества ответов по диапазонам

DOI: https://doi.org/10.60797/IRJ.2026.168.70.4

Обсуждение

Результаты показывают, что качество ответов напрямую зависит от сложности и структуры контекста $C(F(\tau, D))$. Простой промпт даёт наименьшее качество из-за отсутствия внешней информации, тогда как добавление RAG и структурированных данных улучшает результаты, но ограничено шумом и слабой интерпретируемостью неструктурированных источников. Использование структурированных данных повышает стабильность, однако без активных механизмов извлечения их потенциал реализуется частично.

Наибольший прирост качества обеспечивают механизмы вызова инструментов и памяти диалога. Tools позволяют вынести часть вычислений за пределы языковой модели, повышая точность, особенно в задачах с фактами и числами. Память, в свою очередь, обеспечивает накопление релевантного контекста и согласованность ответов в многошаговых сценариях. Это указывает на то, что архитектурные решения (интеграция инструментов и управление состоянием) оказывают более сильное влияние, чем простое увеличение объёма контекста.

Сравнение моделей показывает, что Qwen3.5 обеспечивает более стабильные результаты и существенно более высокую вычислительную эффективность, несмотря на отсутствие языковой адаптации. При сопоставимом среднем качестве она демонстрирует меньшее количество ошибок и лучшее соотношение «качество/время». В целом, результаты подтверждают, что ключевым фактором повышения качества является не масштаб контекста, а эффективность его организации и использования.

Заключение

В нашей работе рассмотрена проблема повышения качества функционирования агентных систем на основе больших языковых моделей за счёт оптимизации контекста, подаваемого на этапе предсказания. Показано, что традиционный статический промпт-инжиниринг, ориентированный на подбор формулировок инструкций, обладает ограниченной эффективностью в прикладных задачах, требующих актуальных знаний, работы с внешними источниками и поддержания связности диалога. Обоснована необходимость перехода к динамической инженерии контекста — подходу, предполагающему структурированное извлечение, фильтрацию и композицию разнородных информационных компонентов (инструкций, внешних знаний, вызовов инструментов, промежуточных рассуждений, памяти и состояния) с учётом специфики информационного домена.

Предложена формальная постановка задачи оптимизации контекста как многокритериальной максимизации ожидаемого вознаграждения при ограничениях на длину контекста, объём доступных данных, вычислительные ресурсы и качество источников. В отличие от существующих работ, фокусирующихся на отдельных аспектах (RAG, память или инструменты), представленная формализация задаёт единую рамку для сравнения и комбинирования различных механизмов управления контекстом.

Полученные результаты подтверждают выдвинутую гипотезу: ключевым фактором эффективности LLM в прикладных задачах является не масштаб контекста сам по себе, а способ его организации, структурирования и интеграции с внешними инструментами и механизмами памяти. Это открывает перспективы для дальнейших исследований в области адаптивной композиции контекста, автоматического выбора наиболее релевантных информационных компонентов в зависимости от типа задачи, а также разработки гибридных архитектур, сочетающих преимущества инструментов, памяти и структурированных знаний в едином фреймворке динамической инженерии контекста.

Конфликт интересов

Не указан.

Рецензия

Все статьи проходят рецензирование. Но рецензент или автор статьи предпочли не публиковать рецензию к этой статье в открытом доступе. Рецензия может быть предоставлена компетентным органам по запросу.

Conflict of Interest

None declared.

Review

All articles are peer-reviewed. But the reviewer or the author of the article chose not to publish a review of this article in the public domain. The review can be provided to the competent authorities upon request.

Список литературы / References

1. Liu P. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing / P. Liu, W. Yuan, J. Fu et al. // ACM computing surveys. — 2023. — № 55. — P. 1–35.

2. Wei J. Chain-of-thought prompting elicits reasoning in large language models / J. Wei, X. Wang, D. Schuurmans et al. // Advances in neural information processing systems. — 2022. — № 35. — P. 24824–24837.

3. Mei L. A Survey of Context Engineering for Large Language Models / L. Mei, J. Yao, Y. Ge et al. // arXiv:2507.13334. — 2025. — URL: https://arxiv.org/abs/2507.13334. (дата обращения: 21.04.26)

4. Lewis P. Retrieval-augmented generation for knowledge-intensive nlp tasks / P. Lewis, E. Perez, A. Piktus et al. // Advances in Neural Information Processing Systems. — 2020. — № 33. — P. 9459–9474.

5. Оболенский Д.М. Использование метода RAG и больших языковых моделей в интеллектуальных образовательных экосистемах / Д.М. Оболенский, В.И. Шевченко // Экономика. Информатика. — 2024. — № 3. — URL: https://cyberleninka.ru/article/n/ispolzovanie-metoda-rag-i-bolshih-yazykovyh-modeley-v-intellektualnyh-obrazovatelnyh-ekosistemah (дата обращения: 21.04.26).

6. Остапенко Г.А. Анализ публикационной активности и научных коллабораций научно-педагогических работников Финансового университета / Г.А. Остапенко, Г.Г. Рожкова, В.Г. Феклин и др. // Цифровые решения и технологии искусственного интеллекта. — 2025. — № 3. — С. 69–76. — DOI: 10.26794/3033-7097-2025-1-3-69-76

7. Гайдамака А.И. Использование GigaCode в деятельности IT-компаний / А.И. Гайдамака, С.Р. Муминова, А.В. Куприянов // Цифровые решения и технологии искусственного интеллекта. — 2025. — № 2. — С. 18–25. — DOI: 10.26794/3033-7097-2025-1-2

8.

9. Волков С.С. Оптимизация контекста больших языковых моделей в высшем техническом образовании / С.С. Волков, С.В. Шалыгин, А.И. Лабинцев // Вестник НИЦ ВА РВСН. — 2025. — № 10. — С. 99–105.

10. Гисин В.Б. Динамическая модель внимания в трансформерах / В.Б. Гисин // Цифровые решения и технологии искусственного интеллекта. — 2025. — № 4. — С. 35–42. — DOI: 10.26794/3033-7097-2025-1-4-35-42

11. Болтачев Э.Ф. Современные методы токенизации текстов в финансовой сфере / Э.Ф. Болтачев, М.П. Фархадов, А.И. Тюляков // Цифровые решения и технологии искусственного интеллекта. — 2025. — № 3. — С. 19–29. — DOI: 10.26794/3033-7097-2025-1-3-19-29

12. Унижаев Н.В. Модель угроз конфиденциальной информации в больших языковых моделях / Н.В. Унижаев. // Цифровая трансформация: тенденции и перспективы : Сборник трудов IV Международной научно-практической конференции; — 4. — Москва: Мир науки, 2025. — С. 992–1003.

13. Швыров В.В. Методы использования больших языковых моделей с поддержкой рассуждений для анализа безопасности программного кода / В.В. Швыров, Д.А. Капустин, Р.Н. Сентяй // Автоматизация в промышленности. — 2026. — № 2. — С. 43–49.

14. Qwen3-8b-ru-i1-GGUF // Hugging Face. — 2026. — URL: https://huggingface.co/mradermacher/Qwen3-8b-ru-i1-GGUF. (дата обращения: 21.04.26)

15. Jin X. Qwen3.5-Omni Technical Report / X. Jin // arXiv:2604.15804. — 2026. — URL: https://arxiv.org/abs/2604.15804. (дата обращения: 21.04.26) doi: 10.48550/arXiv.2604.15804

16. Gu J. A survey on llm-as-a-judge / J. Gu, X. Jiang, Z. Shi et al. // The Innovation. — 2024. — № 1.

Список литературы на английском языке / References in English

1. Liu P. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing / P. Liu, W. Yuan, J. Fu et al. // ACM computing surveys. — 2023. — № 55. — P. 1–35.

2. Wei J. Chain-of-thought prompting elicits reasoning in large language models / J. Wei, X. Wang, D. Schuurmans et al. // Advances in neural information processing systems. — 2022. — № 35. — P. 24824–24837.

3. Mei L. A Survey of Context Engineering for Large Language Models / L. Mei, J. Yao, Y. Ge et al. // arXiv:2507.13334. — 2025. — URL: https://arxiv.org/abs/2507.13334. (accessed: 21.04.26)

4. Lewis P. Retrieval-augmented generation for knowledge-intensive nlp tasks / P. Lewis, E. Perez, A. Piktus et al. // Advances in Neural Information Processing Systems. — 2020. — № 33. — P. 9459–9474.

5. Obolenskii D.M. Ispolzovanie metoda RAG i bolshikh yazikovikh modelei v intellektualnikh obrazovatelnikh ekosistemakh [The Use of the RAG Method and Large Language Models in Intelligent Educational Ecosystems] / D.M. Obolenskii, V.I. Shevchenko // Ekonomika. Informatika [Economics. Informatics]. — 2024. — № 3. — URL: https://cyberleninka.ru/article/n/ispolzovanie-metoda-rag-i-bolshih-yazykovyh-modeley-v-intellektualnyh-obrazovatelnyh-ekosistemah (accessed: 21.04.26). [in Russian]

6. Ostapenko G.A. Analiz publikacionnoj aktivnosti i nauchny'x kollaboracij nauchno-pedagogicheskix rabotnikov Finansovogo universiteta [Analysis of Publication Activity and Scientific Collaborations of Academic Staff of the Financial University] / G.A. Ostapenko, G.G. Rozhkova, V.G. Feklin et al. // Digital Solutions and Artificial Intelligence Technologies. — 2025. — № 3. — P. 69–76. — DOI: 10.26794/3033-7097-2025-1-3-69-76 [in Russian]

7. Gajdamaka A.I. Ispol'zovanie GigaCode v deyatel'nosti IT-kompanij [The Use of GigaCode in the Activities of IT Companies] / A.I. Gajdamaka, S.R. Muminova, A.V. Kupriyanov // Digital Solutions and Artificial Intelligence Technologies. — 2025. — № 2. — P. 18–25. — DOI: 10.26794/3033-7097-2025-1-2 [in Russian]

8.

9. Volkov S.S. Optimizaciya konteksta bol'shix yazy'kovy'x modelej v vy'sshem texnicheskom obrazovanii [Optimization of Large Language Model Context in Higher Technical Education] / S.S. Volkov, S.V. Shaly'gin, A.I. Labincev // Bulletin of the Research Center of the Military Academy of the Strategic Missile Forces. — 2025. — № 10. — P. 99–105. [in Russian]

10. Gisin V.B. Dinamicheskaya model' vnimaniya v transformerax [Dynamic Attention Model in Transformers] / V.B. Gisin // Digital Solutions and Artificial Intelligence Technologies. — 2025. — № 4. — P. 35–42. — DOI: 10.26794/3033-7097-2025-1-4-35-42 [in Russian]

11. Boltachev E'.F. Sovremenny'e metody' tokenizacii tekstov v finansovoj sfere [Modern Methods of Text Tokenization in the Financial Sector] / E'.F. Boltachev, M.P. Farxadov, A.I. Tyulyakov // Digital Solutions and Artificial Intelligence Technologies. — 2025. — № 3. — P. 19–29. — DOI: 10.26794/3033-7097-2025-1-3-19-29 [in Russian]

12. Unizhaev N.V. Model' ugroz konfidencial'noj informacii v bol'shix yazy'kovy'x modelyax [A Threat Model for Confidential Information in Large Language Models] / N.V. Unizhaev. // Digital Transformation: Trends and Prospects: Proceedings of the 4th International Scientific and Practical Conference; — 4. — Moscow: Mir nauki, 2025. — P. 992–1003. [in Russian]

13. Shvy'rov V.V. Metody' ispol'zovaniya bol'shix yazy'kovy'x modelej s podderzhkoj rassuzhdenij dlya analiza bezopasnosti programmnogo koda [Methods of Using Reasoning-Enabled Large Language Models for Software Code Security Analysis] / V.V. Shvy'rov, D.A. Kapustin, R.N. Sentyaj // Automation in Industry. — 2026. — № 2. — P. 43–49. [in Russian]

14. Qwen3-8b-ru-i1-GGUF // Hugging Face. — 2026. — URL: https://huggingface.co/mradermacher/Qwen3-8b-ru-i1-GGUF. (accessed: 21.04.26)

15. Jin X. Qwen3.5-Omni Technical Report / X. Jin // arXiv:2604.15804. — 2026. — URL: https://arxiv.org/abs/2604.15804. (accessed: 21.04.26) doi: 10.48550/arXiv.2604.15804

16. Gu J. A survey on llm-as-a-judge / J. Gu, X. Jiang, Z. Shi et al. // The Innovation. — 2024. — № 1.