

МОДЕЛИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ АГЕНТОВ ДЛЯ АВТОМАТИЗАЦИИ ИНФОРМАЦИОННОГО ПОИСКА

Иванов К.Н., Захарова О.И., Левашкин С.П.

Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ

E-mail: k.ivanov@psuti.ru, o.zaharova@psuti.ru, s.levashkin@psuti.ru

Большие языковые модели продолжают активно развиваться и становятся ключевым инструментом в построении интеллектуальных систем. Их применение уже не ограничивается традиционной генерацией текста и открывает возможности для построения многошаговых рассуждений, интеграции с внешними источниками данных и автоматизации прикладных задач. В данной работе рассматривается моделирование интеллектуальных агентов на основе архитектуры ReAct, реализованной во фреймворке LangChain. Особое внимание уделяется использованию открытых языковых моделей нового поколения (Gemma, Qwen и др.) и их сравнению на основе бенчмарк-теста. Разработанный агент способен осуществлять интеллектуальный поиск информации как в документах, так и в сети, применять внешние инструменты и адаптироваться к различным моделям. Результаты исследования демонстрируют практическую ценность агентного подхода для организации поиска и анализа данных, а также доказывают перспективность использования LLM-агентов в современных прикладных сценариях.

Ключевые слова: искусственный интеллект, большие языковые модели, интеллектуальные агенты, информационный поиск, Python

Введение

Развитие больших языковых моделей (Large Language Model, LLM) существенно изменило подходы к автоматизации процесса решения интеллектуальных задач [1]. Они применяются в поисковых системах, системах поддержки принятия решений и диалоговых интерфейсах, открывая возможности для генерации и анализа информации в самых разных областях [2; 3]. Согласно данным конференции Future of Data-Centric AI [4], более половины организаций планируют внедрение LLM в корпоративные процессы, что подчеркивает их растущее значение для прикладных и научных сценариев (рисунок 1).

How are you planning to use LLMs?

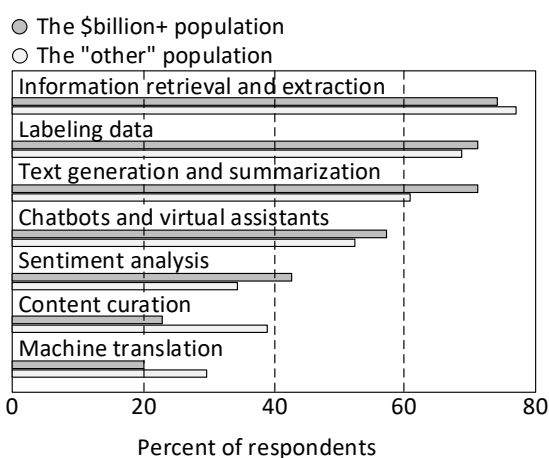


Рисунок 1. Сводная статистика по применению LLM в корпоративном сегменте [4]

При этом, на первый план выходят вопросы приватности, вычислительной эффективности и гибкости развертывания. Использование облачных API-сервисов ограничивает возможности их применения в ряде случаев, связанных с конфиденциальными данными и ограниченностью ресурсов. В этом отношении особый интерес представляют модели нового поколения (Gemma, Qwen и др.), которые обладают сравнительно небольшим размером, способны работать локально и обеспечивают достаточный уровень качества для построения практических решений.

Агентное моделирование, основанное на интеграции LLM с внешними инструментами и возможностью многошаговых рассуждений, рассматривается как один из наиболее перспективных способов повышения качества интеллектуального поиска. Несмотря на широкое распространение данного подхода [5–7], вопрос о том, насколько легкие локальные модели могут эффективно функционировать в агентной архитектуре и обеспечивать результаты, сопоставимые с более крупными системами, остается открытым. В отдельных прикладных исследованиях [8] демонстрировалась ценность подобных подходов, однако систематическая оценка возможностей новых моделей требует детального анализа.

Целью настоящего исследования является создание интеллектуального агента на основе фреймворка LangChain и архитектуры ReAct (Reasoning and Acting), использующего совре-

менные компактные модели для решения задач информационного поиска. Научная новизна работы заключается в демонстрации того, что легкие open-source LLM могут успешно применяться в агентном моделировании, обеспечивая баланс между качеством генерации, автономностью и возможностью локального развертывания. Практическая ценность исследования заключается в том, что такие решения открывают путь к внедрению интеллектуальных агентов в бизнес-среде и в научных приложениях, где особенно важны приватность и контроль над вычислительными ресурсами.

Модель и фреймворк

Для реализации интеллектуального агента в настоящем исследовании использован фреймворк LangChain, обеспечивающий модульность архитектуры, интеграцию внешних инструментов и расширенные возможности работы с агентными системами. LangChain предоставляет встроенные механизмы управления памятью, обработки запросов и подключения языковых моделей, что делает его эффективным инструментом для разработки комплексных систем на основе LLM. Вместе с тем, высокая степень абстракции и сложность документации могут создавать определенные трудности при освоении фреймворка [9].

В качестве основного интерфейса взаимодействия с языковыми моделями используется компонент Ollama (Omni-Layer Learning Language Acquisition Model – модель приобретения языка на основе многослойного обучения), обеспечивающий гибкую настройку моделей под специфические задачи и поддерживающий работу с различными LLM. Для оценки производительности и качества генерации агентной системы в исследовании были выбраны четыре модели, представляющие собой современное поколение локальных и открытых LLM:

- **Gemma 3 27B** – представляет собой модель с 27 миллиардами параметров, оптимизированную для работы на одном графическом процессоре. Она демонстрирует высокую эффективность в решении задач генерации текста, обработки изображений и мультимодальных приложений. Модель поддерживает более 35 языков и способна анализировать как текстовую, так и визуальную информацию, что делает ее особенно полезной для создания интеллектуальных агентов с мультимодальными возможностями;

- **Gemma 3n e4b** – компактная версия семейства Gemma, построенная с использованием техники MatFormer (Matryoshka Transformer) и Per-

Layer Embeddings (PLE), которая обеспечивает селективную активацию подмоделей и снижение расхода памяти. В частности, базовый вес модели содержит до 8 миллиардов параметров, однако благодаря PLE ее эффективная нагрузка может быть эквивалентна 2–4 миллиардам, что делает возможным запуск на устройствах с ограниченной памятью;

- **Qwen3 14B** – компактная модель с 14 миллиардами параметров, входящая в серию Qwen3, которая включает как плотные, так и Mixture-of-Experts (MoE) архитектуры. Модель обеспечивает высокую производительность в задачах кодирования, математического анализа и выполнения агентных операций. Используемая архитектура MoE позволяет эффективно распределять вычислительные ресурсы, активируя только необходимые эксперты, что способствует снижению вычислительных затрат при сохранении качества вывода;

- **Mistral-Small 3.2 24B** – обновленная версия модели с 24 миллиардами параметров. По сравнению с предыдущей версией (Small 3.1), в версии 3.2 улучшены функции следования инструкциям (Instruction Following), снижения бесконечной генерации (Infinite Generations) и устойчивости функции вызова (Function Calling). Модель допускает контекст размером до 128 000 токенов и поддерживает мультимодальность.

Выбор указанных моделей обусловлен совокупностью факторов, обеспечивающих баланс между качеством ответов, скоростью работы и возможностью воспроизведения результатов в локальной вычислительной среде. Во-первых, все модели относятся к категории открытых или локально развертываемых LLM, что позволяет проводить эксперименты вне зависимости от облачных сервисов. Во-вторых, они представляют различные архитектурные подходы (плотные и MoE-структуры), что обеспечивает репрезентативное сравнение алгоритмов при решении агентных задач. В-третьих, эти модели демонстрируют стабильное взаимодействие работы с русскоязычными запросами, поддерживают инструменты function calling и интеграцию с фреймворком LangChain. Наконец, их размер (от 8 до 27 млрд параметров) и требования к ресурсам позволяют эффективно использовать оборудование уровня современного исследовательского сервера без распределенных кластеров.

Тестирование проводилось на сервере, принадлежащем небольшой исследовательской организации, специализирующейся на разработке локальных решений с использованием LLM.

Вычислительная инфраструктура включала двухпроцессорную систему на базе Intel Xeon Gold 6226R (2 × 16 ядер, 64 потока, 2,9 ГГц), 256 ГБ оперативной памяти и графический ускоритель NVIDIA Tesla V100 с 32 ГБ видеопамати. Операционная система – Ubuntu 22,04 LTS (ядро 5,15). Хранилище данных реализовано на LVM-массиве общим объемом около 32 ТБ, что обеспечило достаточный объем для логирования и анализа результатов экспериментов. Локальное развертывание позволило оценить реальные требования к аппаратным ресурсам и возможности воспроизводимости предложенного подхода на оборудовании исследовательского уровня.

Каждая модель была протестирована в унифицированных условиях с использованием LangChain и Ollama для обеспечения сопоставимости результатов.

Для настройки поведения моделей применялись следующие параметры:

- **temperature** – параметр, регулирующий степень креативности генерации. В исследовании использовались значения в диапазоне 0,2–0,4, обеспечивающие баланс между точностью и разнообразием выходных данных;

- **top_p** – порог вероятности выбора токенов, ограничивающий генерацию наиболее вероятными вариантами. Значения варьировались в диапазоне 0,8–0,9, что позволило достичь оптимального соотношения точности и вариативности ответов.

Ранние этапы тестирования показали, что избыточная креативность моделей может снижать точность, особенно при решении задач, требующих строгого соблюдения исходного контекста. В связи с этим предпочтение было отдано более консервативным параметрам настройки, обеспечивающим стабильность и воспроизводимость результатов.

Разработка агента

Агентное моделирование представляет собой метод решения задач с использованием автономных программных сущностей, способных принимать решения на основе поставленных целей и имеющейся информации. В настоящем исследовании интеллектуальные агенты интегрируются с LLM, что позволяет расширить их функциональные возможности за счет взаимодействия с внешними источниками и инструментами. В отличие от традиционных чат-ботов, такие агенты способны выполнять сложные последовательности действий, формировать пошаговые стратегии и адаптироваться к изменяющимся условиям задачи.

Развитие подхода агентного моделирования тесно связано с рядом достижений, включая интегра-

цию плагинов для ChatGPT и создание проектов с открытым исходным кодом, таких как AutoGPT, BabyAGI, AgentGPT и LangChain. Эти решения позволяют агентам строить пошаговые планы, использовать дополнительные источники данных и повышать надежность своих решений, снижая влияние галлюцинаций, присущих языковым моделям.

В настоящем исследовании агент реализован на базе фреймворка LangChain с использованием архитектуры ReAct, обеспечивающей синергию между рассуждением и действием. Агент анализирует входные данные, формирует обоснованные действия на их основе и динамически взаимодействует с пользователем и внешними инструментами.

Инструменты служат для расширения возможностей агента за пределами встроенных знаний LLM [10]. В LangChain предусмотрены компоненты для работы с внешними источниками данных, включая подключение к Интернету и обработку локальных документов. Для поиска информации в сети может быть использован, например, GoogleSerperAPIWrapper. Для работы с локальными файлами применяются загрузчики, поддерживающие форматы CSV, DOCX, PDF и TXT. Эти инструменты позволяют агенту эффективно обрабатывать отчеты, научные публикации и документацию (схема работы представлена на рисунке 2).

Для работы с HTML-страницами разработаны специализированные решения на базе Selenium и BeautifulSoup, обеспечивающие фильтрацию текста и формирование цепочек «вопрос–ответ» (схема представлена на рисунке 3).

Разработанная авторами архитектура интеллектуального агента (рисунки 4 и 5) представляет собой развитие подхода ReAct с расширением механизма взаимодействия между этапами рассуждения и действия. В отличие от типовых решений LangChain-агентов, реализована разделенная структура пайплайнов, включающая независимые модули рассуждения, инструментальной обработки и векторного поиска. Такая декомпозиция позволяет изолировать операции логического вывода от внешних обращений, что повышает воспроизводимость и устойчивость генерации.

Существенным отличием предложенного решения является использование двухуровневой системы памяти: краткосрочная память обеспечивает непрерывность диалога, а долгосрочная память сохраняет результаты обращений к FAISS-базам документов и HTML-страниц. Это позволило обеспечить контекстное накопление знаний без необходимости постоянного повторного извлечения данных.

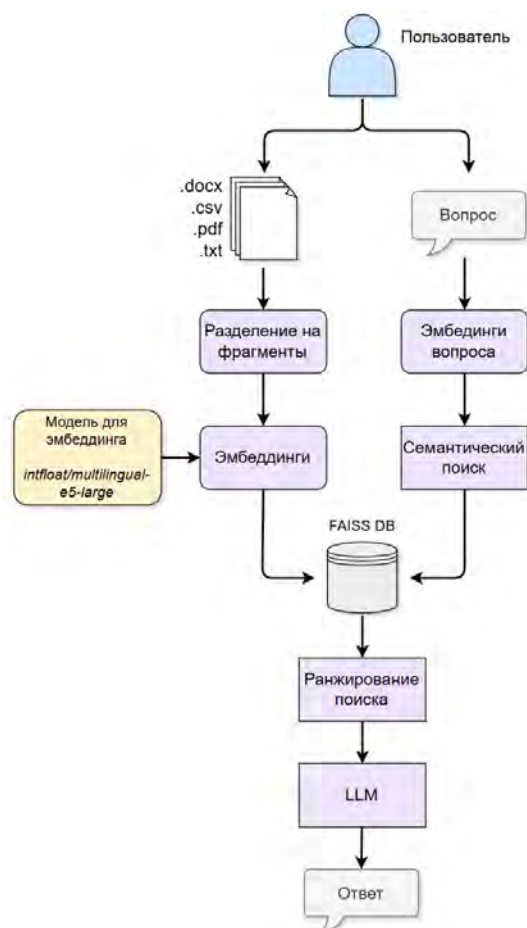


Рисунок 2. Схема взаимодействия пользователя с инструментальным функционалом интеллектуального агента для формирования ответов на основе загруженных документов

В архитектуру также включен гибридный RAG-модуль, объединяющий локальные источники (DocGPT, HtmlGPT) с онлайн-инструментами (Wikipedia API, Google API), что обеспечивает баланс между скоростью локального поиска и актуальностью сетевых данных. Такая интеграция дает возможность агенту работать как в полностью офлайн-среде, так и при подключении к Интернету.

Представленная схема отражает вклад авторов в развитие архитектур агентных систем в части организации взаимодействия между рассуждением, памятью и инструментами, а также адаптации многоисточникового поиска под локальные вычислительные условия. Это обеспечивает возможность масштабирования решения и его применения в исследовательских и производственных сценариях.

Память обеспечивает сохранение контекста предыдущих взаимодействий, устраняя разрывы в диалоге и повышая качество ответов. В локальных решениях память позволяет обходить огра-

ничения на длину контекстного окна и экономить вычислительные ресурсы. Для рассматриваемого агента оптимальным выбором является тип памяти ConversationBufferWindowMemory, который сохраняет последние k диалогов, обеспечивая баланс между объемом памяти и производительностью.

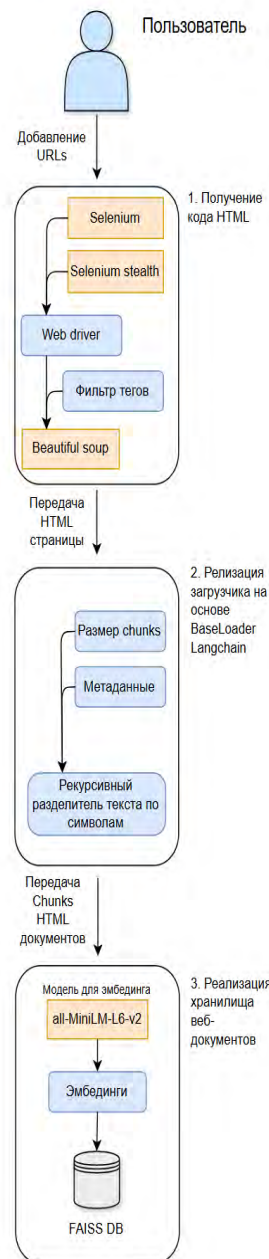


Рисунок 3. Схема разработки инструмента для анализа HTML-страниц и поиска ответов на вопросы

Подсказка (prompt) задает структуру взаимодействия агента с пользователем. Она включает описание инструментов, сценарии поведения и примеры их использования. Корректная формулировка подсказки повышает точность выбора инструментов, корректность обработки запросов и эффективность использования контекста модели.

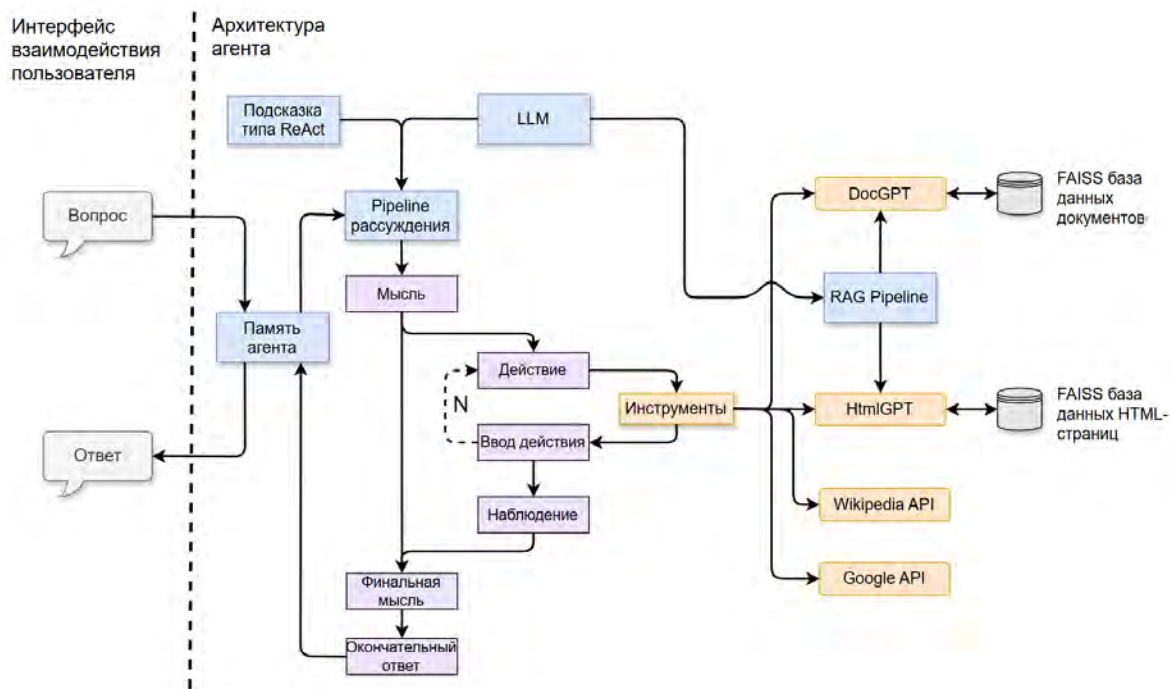


Рисунок 4. Общая схема работы агента

```

> Entering new AgentExecutor chain...
...
Question: Какие инструменты у тебя есть?
Thought: Следует ли использовать инструмент для этого? Нет
Final Thought: Теперь у меня есть окончательный ответ.
Final Answer: У меня есть доступ к Google Search, DocRAG и HtmlRAG.
...

> Finished chain.

AI: {'input': 'Какие инструменты у тебя есть?', 'chat_history': 'Human: Привет!\nAI: Привет! 🌟 Чем я могу вам помочь? \n'''\nHuman: Что ты умеешь? \nAI: Я полезный, уважительный и надежный AI-помощник. Я могу отвечать на ваши вопросы, используя свои знания и доступные мне инструменты. \n'''\n', 'output': 'У меня есть доступ к Google Search, DocRAG и HtmlRAG. \n'''\n}

User: Хорошо. DocRAG - какие ключевые слова в тексте?

> Entering new AgentExecutor chain...
...
Question: Хорошо. DocRAG - какие ключевые слова в тексте?
Thought: Следует ли использовать инструмент для этого? Да, DocRAG может помочь найти ключевые слова в тексте.
Action: DocRAG
Action Input: Пожалуйста, найдите ключевые слова в тексте.

> Entering new RetrievalQA chain...

> Finished chain.

Observation: The keywords are:
* экосистемный анализ (ecosystem analysis)
* экологические и природно-технические системы (ecological and natural-technogenic systems)
* экологические функции (ecological functions)
* информация (information)
* природные ресурсы (natural resources)

Теперь я знаю ответ на вопрос, основываясь на наблюдении.
Final Answer: Ключевые слова в тексте: экосистемный анализ, экологические и природно-технические системы, экологические функции, информация, природные ресурсы.
...

```

Рисунок 5. Пример работы разработанного агента. Выполнение поискового запроса с использованием инструментов

Общая логика работы агента включает следующие принципы:

1. Анализ запроса – агент определяет цель действия и сопоставляет ее с доступными инструментами и знаниями LLM.
2. Принятие решения – формирование последовательности действий на основе анализа запроса и доступных ресурсов.
3. Выполнение действий – обращение к инструментам, обработка документов или интернет-ресурсов.
4. Формирование ответа – объединение данных модели и инструментов для предоставления пользователю корректного результата.

5. Сохранение контекста – обновление памяти, обеспечивающее непрерывность диалога и возможность использования предыдущего опыта в будущих запросах.

Тестирование и результаты

Экспериментальная часть исследования была направлена на проверку применимости методологии в целом и оценку различных вариантов ее реализации. В частности, сравнивались методы прямого извлечения, специализированные модели и многоступенчатые схемы. Такой подход позволил выявить преимущества и ограничения каждого из них в рамках общей схемы построения графов знаний.

Методологической основой выступила ранее описанная архитектура ReAct, которая сочетает рассуждение и выполнение действий. В рамках тестирования агенты могли как самостоятельно генерировать ответы, так и обращаться к внешним инструментам для поиска дополнительного контекста в целях формирования ответа. Такой подход позволил оценить не только «внутренние» возможности моделей, но и их способность к интеграции инструментального функционала.

Данные для исследования

С целью оценки производительности интеллектуальных агентов был сформирован бенчмарк, включающий пять типов задач, различающихся по когнитивной сложности и требованиям к рассуждениям:

- GSM8K (20 вопросов выборки из полного набора) – математические задачи начального уровня сложности, требующие пошагового решения и демонстрирующие способность моделей к формальным рассуждениям [11];
- HotpotQA-easy (20 вопросов) – простые вопросы, требующие поиска фактов в одном или двух источниках;
- HotpotQA-medium (20 вопросов) – вопросы средней сложности, где необходимо связать несколько фактов, но без значительной глубины многошагового анализа;
- HotpotQA-hard (20 вопросов) – задачи, предполагающие глубокие многошаговые рассуждения и интеграцию информации из нескольких источников [12];
- GAIA (20 вопросов) – комплексный набор для оценки возможностей «универсальных ассистентов», включающий разнотипные задачи: логические, мультимодальные и инструментальные [13].

Общая оценка производительности

Для комплексной оценки эффективности работы каждой модели был введен интегральный показатель Score, который агрегирует значения ключевых метрик и характеристик агента. Формально он задается следующим образом:

$$Score = 0,4 \cdot \overline{Acc} + 0,2 \cdot \overline{Comp} + 0,2 \cdot \overline{Rel} + \\ + 0,1 \cdot \overline{Tool}_{use} + 0,1 \cdot \overline{React}_{comp},$$

где $\overline{Acc}$ – среднее значение точности (Accuracy), отражающее корректность воспроизведения структурных элементов ответа и ключевых слов. На эту метрику приходится наибольший вес (40%), так как именно точность напрямую связана с правильностью решения задачи;

$\overline{Comp}$ – среднее значение полноты

(Completeness), характеризующее степень охвата всех аспектов эталонного ответа. Вклад 20% обеспечивает баланс между необходимостью содержательной полноты и предотвращением переоценки избыточных ответов;

$\overline{Rel}$ – среднее значение релевантности (Relevance), которое фиксирует степень соответствия ответа поставленной задаче и корректность выбранной стратегии рассуждений. Вклад также равен 20%;

$\overline{Tool}_{use}$ – показатель эффективности использования инструментов. Он учитывает, насколько адекватно и результативно агент применяет внешние средства (например, доступ к Интернету или обработку документов) при решении задач. Вес 10% отражает роль инструментов как вспомогательного компонента архитектуры ReAct;

$\overline{React}_{comp}$ – показатель соблюдения архитектуры ReAct, выражающий способность агента чередовать этапы рассуждения и действия. Этот критерий также имеет вес 10%, так как ReAct-парадигма является ключевой методологической основой построения агента.

Выбор указанных коэффициентов основан на трех принципах:

1. Смысловая значимость метрик – Accuracy имеет первостепенное значение, Completeness и Relevance обеспечивают содержательную полноту, а $\overline{Tool}_{use}$ и $\overline{React}_{comp}$ являются вспомогательными компонентами, влияющими на качество вторично.

2. Экспертная эвристика – веса отражают опыт авторов в оценке поведения ReAct-агентов, учитывающий относительное влияние каждой метрики на итоговую адекватность ответов.

3. Эмпирическая верификация – проведены тесты с альтернативными комбинациями весов, которые показали незначительное изменение ранжирования моделей и интегральной оценки. Выбранные коэффициенты обеспечивают наибольшую согласованность Score с качественными наблюдениями по поведению агентов и наиболее корректно отражают ключевые аспекты их работы.

На рисунке 6 представлены результаты анализа интегрального показателя. Qwen3 14B продемонстрировала наилучший результат, обеспечив оптимальный баланс между качеством рассуждений, эффективностью использования инструментов и устойчивостью к аномалиям. Напротив, модель Gemma 3n:e4b получила минимальный интегральный балл, что связано с ее меньшим размером и ограниченной способностью выполнять многошаговые рассуждения в сложных сценариях:

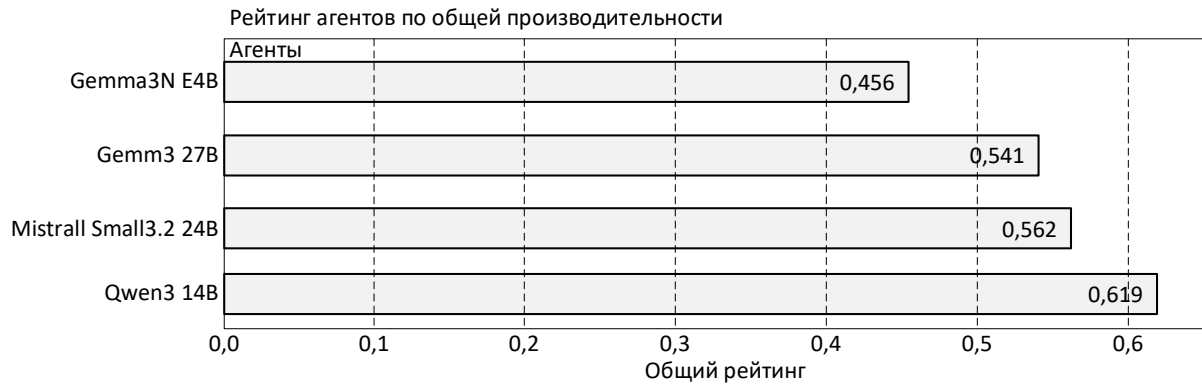


Рисунок 6. Интегральный рейтинг агентов по производительности

Качество ответов

Качество сгенерированных ответов оценивалось по четырем основным метрикам: точность (Accuracy), полнота (Completeness), релевантность (Relevance) и семантическое сходство (Semantic Similarity). Каждая метрика учитывает различные аспекты качества вывода и позволяет всесторонне охарактеризовать поведение модели.

Точность – отражает долю правильно определенных элементов в ответе с учетом совпадения ключевых слов. Формально:

$$Accuracy = \frac{C_{correct}}{C_{total}} \cdot 0,8 + \frac{K_{resp}}{K_{gold}} \cdot 0,2,$$

где $C_{correct}$ – число правильно определенных элементов;

C_{total} – общее число элементов;

K_{resp} – количество совпавших ключевых слов в ответе;

K_{gold} – число ключевых слов в эталоне.

Таким образом, 80% веса метрики определяется фактической правильностью структурных элементов ответа, а 20% – совпадением ключевых слов. Такой баланс позволяет одновременно учитывать как структурную корректность вывода, так и сохранение ключевых понятий.

Полнота – фиксирует степень покрытия ответа относительно ожидаемой длины и числа аспектов задачи. Она определяется как:

$$Completeness = \min \left(1, 0, \frac{L_{resp}}{L_{exp}} \cdot 0,3 + \frac{A_{cov}}{A_{tot}} \cdot 0,7 \right),$$

где L_{resp} – длина ответа;

L_{exp} – ожидаемая длина;

A_{cov} – число покрытых аспектов;

A_{tot} – общее число аспектов.

Здесь основной вес (70%) отводится аспектному покрытию, что подчеркивает важность полноты изложения, а дополнительный вес (30%) – длине ответа. Верхняя граница функции ограни-

чена значением 1,0, что исключает возможность «переполнения» метрики при чрезмерно длинных ответах.

Релевантность – отражает степень соответствия ответа поставленной задаче, как на уровне используемой лексики, так и на уровне совпадения с типом задачи:

$$Relevance = \frac{W_{rel}}{W_{all}} \cdot 0,6 + \frac{T_{match}}{T_{max}} \cdot 0,4,$$

где W_{rel} – релевантные слова;

W_{all} – общее число слов;

T_{match} – соответствие типу задач;

T_{max} – максимально возможное значение.

Таким образом, 60% веса приходится на использование релевантной лексики, а 40% – на соответствие типу задачи. Это позволяет сбалансировать метрику между лексическим качеством и правильностью выбора стратегии ответа.

Семантическое сходство интегрирует оценку глобальной смысловой близости между ответом модели и эталоном с проверкой совпадения ключевых слов:

$$Sim = 0,7 \cos(A, B) + 0,3 \cdot \frac{|K_{resp} \cap K_{gold}|}{K_{gold}},$$

где $\cos(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$ – конусное сходство между векторными представлениями ответа и эталона;

K_{resp} – множество ключевых слов, выделенных из ответа модели;

K_{gold} – множество ключевых слов эталонного ответа.

Данная метрика позволяет одновременно учитывать глобальную семантическую близость текстов (70% веса) и точность воспроизведения ключевых понятий (30% веса), что критично при тестировании задач с требованием многошагового рассуждения и формальной правильности.

На рисунке 7 сравнены ключевые метрики качества ответов для каждой из моделей – точность (Accuracy), полнота (Completeness), релевантность (Relevance), ясность (Clarity) и семантическое сходство (Semantic Similarity). Каждый столбец/ось соответствует усредненному значению метрики по 100 вопросам бенчмарка.

Различия отражают сочетание архитектурных особенностей и способности модели поддерживать последовательность рассуждений: высокий балл по Accuracy/Completeness у Qwen3 означает, что модель чаще генерирует корректные структурированные ответы и покрывает ожидаемые аспекты задачи; у Gemma 3n:e4b наблюдается дефицит контекстной глубины и охвата аспектов.

Рисунок 8 визуализирует матрицу производительности агентов по типам задач. Ключевые наблюдения:

- GAIA оказалась наиболее сложным набором (средняя точность – 0,34). Все модели испытывают затруднения, наихудший результат показала Gemma 3:27B (0,23);

- HotpotQA-easy (средняя точность – 0,605) демонстрирует, что большинство агентов успешно справляются с простыми фактологическими вопросами, особенно Qwen 3:14B (0,8);

- GSM8K (0,62) выявил специализацию Gemma 3:27B (0,76), тогда как Mistral показал низкий результат (0,31);

- HotpotQA-medium/hard (0,690–0,708) подтвердили преимущество крупных моделей, в первую очередь Qwen 3:14B и Mistral (обе 0,8), что отражает их способность к многошаговым рассуждениям.

На рисунке 9 представлено распределение качества ответов протестированных моделей. Распределение качества ответов подчеркивает как различия между моделями по уровню когнитивной сложности, так и специфику их применения. Если Qwen 3 14B можно рассматривать как универсального кандидата для агентных архитектур, то Gemma 3n:e4b остается скорее демонстрацией возможностей компактных решений, требующих дальнейшего совершенствования.

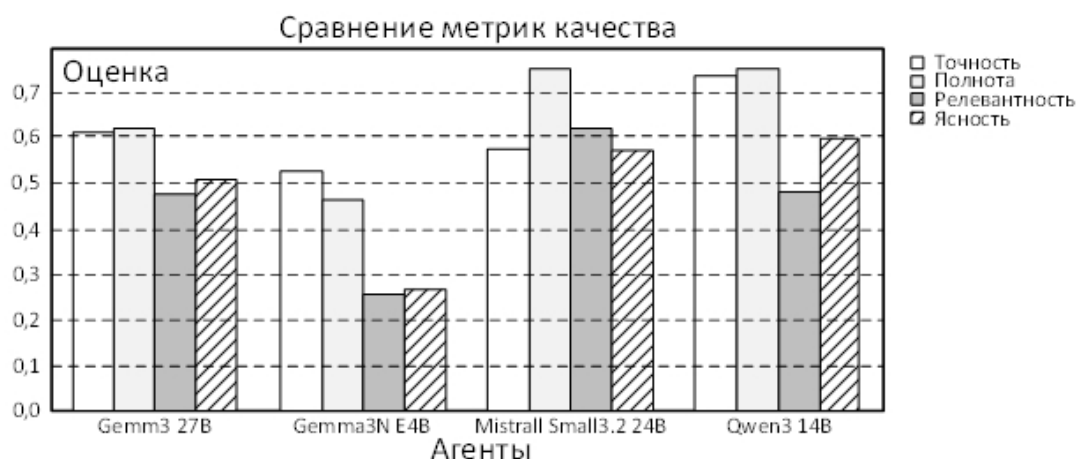


Рисунок 7. Сравнение метрик качества агентов

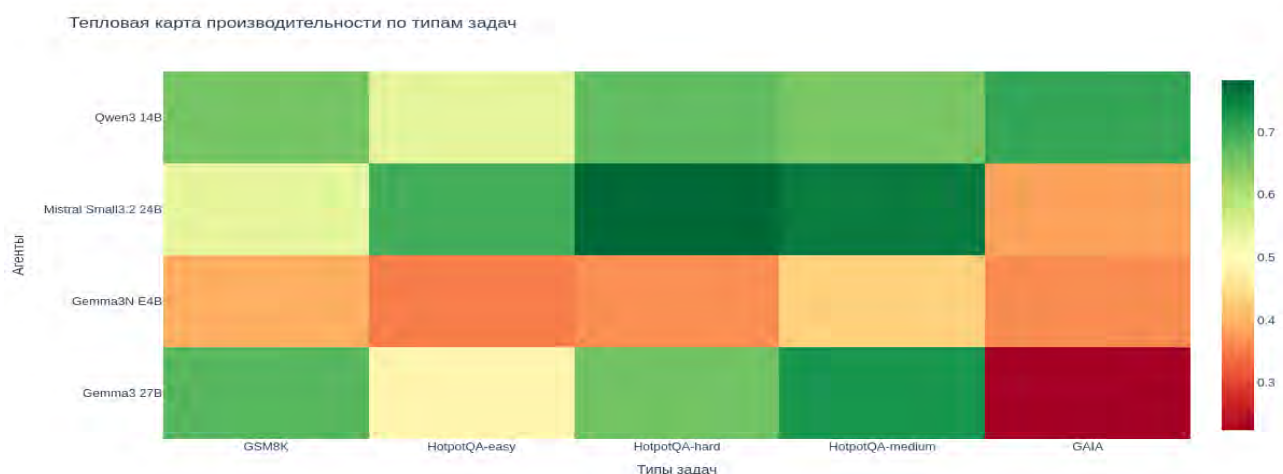


Рисунок 8. Тепловая карта производительности агентов по типам задач

На рисунке 10 представлен график распределения использования инструментов агентами в процессе прохождения теста. Ключевые наблюдения:

- Qwen 3:14B чаще всего использует инструменты в задачах HotpotQA (search-heavy), что коррелирует с ее высокой точностью по этим типам вопросов;
- Gemma 3n:e4b – минимальная частота использования инструментов, даже там, где поиск был бы полезен;
- Mistral и Gemma 3:27B используют инструменты умеренно; чаще в GAIA/HotpotQA.

На рисунке 11 представлена визуализация коэффициента корреляции Пирсона. Частота использования инструментов отражает стратегию агента: активное привлечение внешних ресурсов повышает вероятность корректного ответа в сценариях с не-

достатком встроенных знаний модели. Однако частота сама по себе не гарантирует эффективность – важна корректность и релевантность применения.

Основные результаты комплексной оценки моделей приведены в таблицах 1 и 2. Таблица 1 содержит сводные показатели производительности агентов по ключевым метрикам (точность, полнота, релевантность, ясность, семантическое сходство, использование инструментов и соответствие архитектуре ReAct), что позволяет наглядно сравнить интегральный результат Score и отдельные составляющие. Таблица 2 демонстрирует эффективность использования ресурсов VRAM (Video Random Access Memory – видеопамять с произвольным доступом) относительно количества параметров модели, что позволяет оценить соотношение вычислительной нагрузки и производительности.

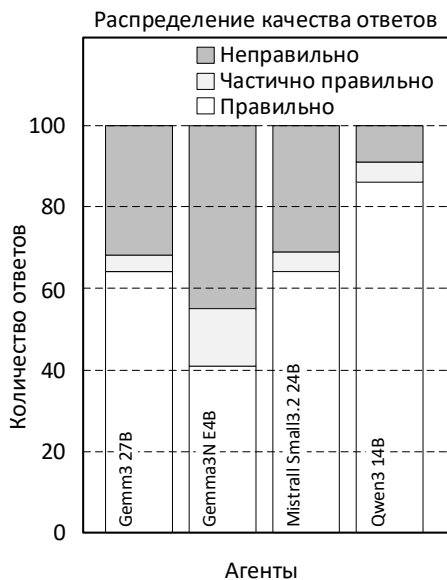


Рисунок 9. Распределение качества ответов в сравнении с эталонным ответов в собранном тестовом наборе

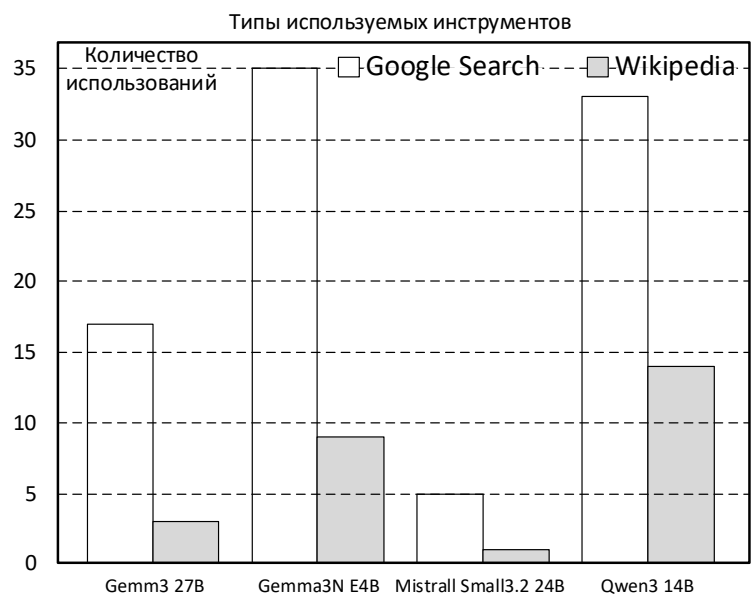


Рисунок 10. Частота использования инструментов с распределением по их типу

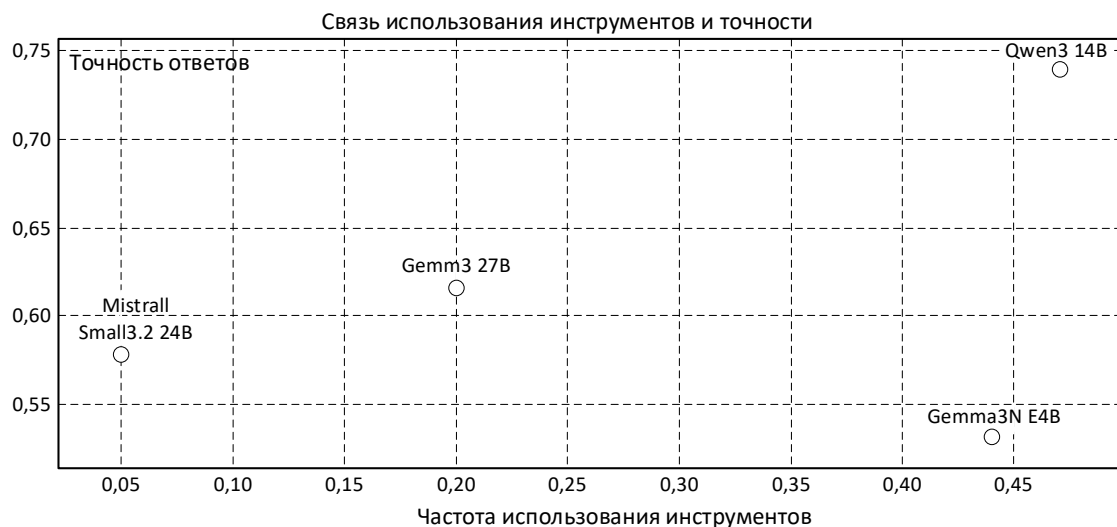


Рисунок 11. Корреляция использования инструментов и точности ответов

Таблица 1. Сводная таблица производительности агентов

	Qwen3 14b	Mistral-small 3.2 24b	Gemma 3 27b	Gemma 3n e4b	Среднее
Общий рейтинг	0,619	0,558	0,545	0,456	0,544
Точность	0,742	0,681	0,654	0,521	0,650
Полнота	0,678	0,612	0,589	0,523	0,601
Релевантность	0,654	0,587	0,554	0,489	0,571
Ясность	0,623	0,587	0,554	0,489	0,563
Семантическое сходство	0,623	0,587	0,554	0,489	0,563
Использование инструментов	0,342	0,298	0,267	0,189	0,274
Соответствие ReAct	0,678	0,612	0,589	0,523	0,601

Таблица 2. Сводные данные использования VRAM

	Qwen3 14b	Mistral-small 3.2 24b	Gemma 3 27b	Gemma 3n e4b
Количество параметров модели	14	24	27	4
Пиковое использование VRAM (Gb)	16,8	20,1	24,5	8,2
Эффективность Количество параметров / VRAM	0,83	1,19	1,10	0,49

Проблемы и вызовы

Проведенное исследование позволило выявить ряд ключевых вызовов, сопровождающих использование современных языковых моделей в агентных архитектурах. Эти вызовы следует рассматривать не только как ограничения, но и как перспективные направления для совершенствования подходов и технологий.

Одним из наиболее значимых факторов остаются требования к вычислительным ресурсам. Даже модели среднего и большого масштаба (24–27B) демонстрируют высокое потребление видеопамяти, достигая в пиковых нагрузках более 24 GB VRAM. Такая особенность ограничивает спектр устройств для локального развертывания и требует применения квантования, распределенных вычислений и адаптивных методов оптимизации. При этом подобные стратегии открывают возможность поиска баланса между вычислительной эффективностью и качеством вывода, что стимулирует развитие специализированных техник настройки и внедрение архитектур типа Mixture-of-Experts.

Другим важным вызовом является ограничение контекстного окна. Задачи, связанные с обработкой длинных документов и сложных цепочек рассуждений, требуют разработки более изощренных механизмов работы с текстом. В этом контексте особую ценность приобретает использование Retrieval-Augmented Generation (RAG), интеграция внешних индексов и гибридных стратегий суммаризации, позволяющих сохранять когерентность ответа при работе с объемными данными.

Особое внимание заслуживает проблема интеграции инструментов. Несмотря на то, что использование внешних ресурсов повышает качество решений, эффективность этого процесса во многом определяется корректностью оркестрации. Некорректная интерпретация результатов поиска или анализа документов может снижать достоверность вывода. Решение видится в явной спецификации инструментов внутри подсказок, создании специализированных парсеров и включении модулей автоматической верификации.

Чувствительность моделей к структуре подсказки также остается предметом анализа. Избыточные и перегруженные инструкциями промпты способны снижать эффективность генерации. В этой связи представляют интерес компактные и модульные шаблоны подсказок, динамическое контекстирование и стратегии адаптивного включения примеров, которые позволяют сохранять точность при экономии ресурсов.

Еще одним направлением, требующим дальнейшей проработки, является устойчивость рассуждений. Зафиксированные случаи циклических или избыточных шагов в ReAct-процессах подчеркивают необходимость внедрения ограничителей числа итераций, а также методов самосогласованности и альтернативной проверки гипотез, повышающих стабильность работы агента.

Наконец, важным вызовом остается обеспечение воспроизводимости и конфиденциальности. Различия в версиях моделей и параметрах запуска, а также использование облачных инструментов при работе с потенциально чувствительными

данными требуют более жесткой стандартизации и документирования вычислительных окружений. При этом развитие локальных решений и использование методов защиты информации позволяют сохранять контроль над данными без снижения качества вычислений.

В совокупности перечисленные вызовы не снижают значимости проведенного исследования, но очерчивают ключевые направления дальнейших разработок. Они указывают на то, что будущее развитие агентных систем будет связано с совершенствованием методов оптимизации, расширением механизмов взаимодействия.

Заключение

В данной работе был реализован и всесторонне оценен интеллектуальный агент в парадигме ReAct, основанный на четырех современных языковых моделях: Qwen3 14B, Mistral-Small 3.2 24B, Gemma 3 27B и Gemma 3n e4b. Для анализа были использованы несколько типов задач различной сложности (GSM8K, HotpotQA-easy/medium/hard, GAIA), что позволило оценить как качество ответов (Accuracy, Completeness, Relevance, Semantic Similarity), так и способность агентов к корректной инструментальной интеграции и наблюдению логики ReAct.

Результаты исследования показали, что Qwen3 14B является наиболее универсальной моделью, обеспечивающей высокий уровень точности и семантической согласованности при эффективном использовании инструментов. Mistral-Small 3.2 24B проявила лучшие качества в задачах с многошаговыми рассуждениями и может рассматриваться как перспективный выбор для сценариев, требующих глубокого анализа и интеграции информации. Gemma 3 27B продемонстрировала устойчивые показатели и сбалансированный профиль, что делает ее применимой в мультимодальных приложениях при наличии достаточных вычислительных ресурсов. В то же время компактная модель Gemma 3n e4b подтвердила потенциал легких решений, однако ее текущая эффективность остается ограниченной и требует дальнейшей оптимизации.

С практической точки зрения ключевым преимуществом протестированных систем является возможность их локального развертывания. В условиях возрастающего внимания к вопросам конфиденциальности данных и контроля вычислительных ресурсов это открывает перспективы внедрения агентных систем в корпоративные и исследовательские процессы, где использование облачных API может быть нежелательным. Такие

решения особенно востребованы в сценариях, связанных с обработкой чувствительных данных, поддержкой принятия решений, интеллектуальным поиском и автоматизацией аналитики.

Перспективные направления дальнейших исследований включают: совершенствование инструментальной оркестрации и методов контроля качества рассуждений (включая механизмы самосогласованности и верификаторы наблюдений), развитие методов балансировки между энергозатратностью и качеством (Distillation, Offloading, Mixture-of-Experts), а также расширение области применения за счет мультимодальных сценариев и предметно-ориентированных задач в бизнесе, медицине, инженерии и других областях.

Таким образом, результаты проведенного исследования подтверждают высокую практическую ценность использования локально развертываемых LLM-агентов и подчеркивают, что направление разработки агентных систем остается одним из наиболее перспективных в области интеллектуального анализа данных и автоматизации прикладных процессов.

Литература

1. Иванов К.Н., Захарова О.И. Обработка естественного языка. Применение языковых моделей // Актуальные проблемы информатики, радиотехники и связи: материалы XXX Российской научно-технической конференции. Самара: Поволжский государственный университет телекоммуникаций и информатики, 2023. С. 155–156.
2. Иванов К.Н., Кушуков С.В. Обработка табличных данных с применением pandasai // Проблемы техники и технологий в телекоммуникациях: материалы XXVI Международной научно-технической конференции. Самара: Поволжский государственный университет телекоммуникаций и информатики, 2024. С. 246–247.
3. Иванов К.Н., Кушуков С.В. Русскоязычные LLM: сравнительный анализ и перспективы применения в проектах // Актуальные проблемы информатики, радиотехники и связи: материалы XXXI Российской научно-технической конференции. Самара: Поволжский государственный университет телекоммуникаций и информатики, 2024. С. 240–241.
4. Snorkel. LLMs high priority for enterprise data science, but concerns remain. URL: <https://snorkel.ai/poll-data-llm-high-priority-enterprise-data-science-concerns-remain/> (дата обращения: 06.08.2025).

5. Training language models to follow instructions with human feedback / L. Ouyang [et al.]. URL: https://cdn.openai.com/papers/Training_language_models_to_follow_instructions_with_human_feedback.pdf (дата обращения: 06.08.2025).
 6. Enabling intelligent interactions between an agent and an LLM: A reinforcement learning approach / B. Hu [et al.]. URL: <https://arxiv.org/pdf/2306.03604> (дата обращения: 06.08.2025).
 7. AutoGen: Enabling next-gen LLM applications via multi-agent conversation / Q. Wu [et al.]. URL: <https://arxiv.org/pdf/2308.08155> (дата обращения: 06.08.2025).
 8. Pandya K., Holia M. Automating customer service using LangChain: Building custom open-source GPT chatbot for organizations. URL: <https://arxiv.org/pdf/2310.05421> (дата обращения: 06.08.2025).
 9. The problem with LangChain. URL: <https://www.pelayoarbues.com/literature-notes/Articles/The-Problem-With-LangChain/> (дата обращения: 06.08.2025).
 10. Chain-of-thought prompting elicits reasoning in large language models / J. Wei [et al.]. URL: <https://arxiv.org/pdf/2201.11903> (дата обращения: 06.08.2025).
 11. MR-GSM8K: A meta-reasoning benchmark for large language model evaluation / Z. Zeng [et al.]. URL: <https://arxiv.org/pdf/2312.17080> (дата обращения: 06.08.2025).
 12. HotpotQA: A dataset for diverse, explainable multi-hop question answering / Z. Yang [et al.]. URL: <https://arxiv.org/pdf/1809.09600> (дата обращения: 06.08.2025).
 13. GAIA: A global, multi-modal, multi-scale vision-language dataset for remote sensing image analysis / A. Zavras [et al.]. URL: <https://arxiv.org/pdf/2502.09598> (дата обращения: 06.08.2025).
- Дата поступления 20.08.2025
Дата принятия 20.09.2025

Иванов Константин Николаевич, аспирант кафедры информационных систем и технологий (ИСТ) Поволжского государственного университета телекоммуникаций и информатики (ПГУТИ). 443010, Российская Федерация, г. Самара, ул. Л. Толстого, 23. Тел. +7 937 995-66-88. E-mail: k.ivanov@psuti.ru

Захарова Оксана Игоревна, к.т.н., доцент кафедры ИСТ, заместитель заведующего научно-исследовательской лабораторией искусственного интеллекта (НИЛ ИИ) ПГУТИ. 443010, Российская Федерация, г. Самара, ул. Л. Толстого, 23. Тел. +7 906 343-25-21. E-mail: o.zaharova@psuti.ru

Левашкин Сергей Павлович, к.ф.-м.н., доцент кафедры ИСТ, заведующий НИЛ ИИ ПГУТИ. 443010, Российская Федерация, г. Самара, ул. Л. Толстого, 23. Тел. +7 917 102-08-98. E-mail: s.levashkin@psuti.ru

UDC 004.82: 004.92
DOI: 10.18469/ikt.2025.23.3.13

MODELING INTELLIGENT AGENTS FOR INFORMATION RETRIEVAL AUTOMATION

Ivanov K.N., Zakharova O.I., Levashkin S.P.

Povolzhskiy State University of Telecommunications and Informatics, Samara, Russian Federation

E-mail: k.ivanov@psuti.ru, o.zaharova@psuti.ru, s.levashkin@psuti.ru

Large Language Models continue to evolve rapidly and have become a key tool in the development of intelligent systems. Their applications extend beyond traditional text generation, enabling multi-step reasoning, integration with external data sources, and automation of applied tasks. This paper explores the modeling of intelligent agents based on the ReAct architecture implemented within the LangChain framework. Special attention is given to the use of new-generation open-source language models (such as Gemma, Qwen, and others) and their comparison through benchmark testing. The developed agent is capable of performing intelligent information retrieval across documents and the web, utilizing external tools, and adapting to various models. The results of the study demonstrate the practical value of the agent-based approach for organizing data search and analysis, and emphasize the promising potential of LLM-based agents in modern applied scenarios.

Keywords: artificial intelligence, large language models, intelligent agents, information retrieval, Python

Ivanov Konstantin Nikolaevich, Povolzhskiy State University of Telecommunications and Informatics, 23, L. Tolstoy Street, Samara, 443010, Russian Federation; PhD Student of Information Systems and Technologies Department. Tel. +7 937 995-66-88. E-mail: k.ivanov@psuti.ru

Zakharova Oksana Igorevna, Povolzhskiy State University of Telecommunications and Informatics, 23, L. Tolstoy Street, Samara, 443010, Russian Federation; Associated Professor of Information Systems and Technologies Department, Deputy Head of the Scientific Research Laboratory of Artificial Intelligence, PhD in Technical Science. Tel. +7 906 343-25-21. E-mail: o.zaharova@psuti.ru

Levashkin Sergey Pavlovich, Povolzhskiy State University of Telecommunications and Informatics, 23, L. Tolstoy Street, Samara, 443010, Russian Federation; Associated Professor of Information Systems and Technologies Department, Head of the Artificial Intelligence Research Laboratory, PhD in Physics and Mathematics Sciences. Tel. +7 917 102-08-98. E-mail: s.levashkin@psuti.ru

References

1. Ivanov K.N., Zakharova O.I. Natural language processing. Application of language models. *Aktualnye problemy informatiki, radiotekhniki i svyazi: materialy XXX Rossijskoj nauchno-tekhnicheskoy konferencii*. Samara: Povolzhskij gosudarstvennyj universitet telekommunikacij i informatiki, 2023, pp. 155–156. (In Russ.)
2. Ivanov K.N., Kushukov S.V. Processing of tabular data using pandasai. *Problemy tekhniki i tekhnologij v telekommunikacijah: materialy XXVI Mezhdunarodnoj nauchno-tekhnicheskoy konferencii*. Samara: Povolzhskij gosudarstvennyj universitet telekommunikacij i informatiki, 2024, pp. 246–247. (In Russ.)
3. Ivanov K.N., Kushukov S.V. Russian-language LLMs: comparative analysis and application prospects in projects. *Aktualnye problemy informatiki, radiotekhniki i svyazi: materialy XXXI Rossijskoj nauchno-tekhnicheskoy konferencii*. Samara: Povolzhskij gosudarstvennyj universitet telekommunikacij i informatiki, 2024, pp. 240–241. (In Russ.)
4. Snorkel. LLMs high priority for enterprise data science, but concerns remain. URL: <https://snorkel.ai/poll-data-llm-high-priority-enterprise-data-science-concerns-remain/> (accessed: 06.08.2025).
5. Ouyang L. et al. Training language models to follow instructions with human feedback. URL: https://cdn.openai.com/papers/Training_language_models_to_follow_instructions_with_human_feedback.pdf (accessed: 06.08.2025).
6. Hu B. et al. Enabling intelligent interactions between an agent and an LLM: A reinforcement learning approach. URL: <https://arxiv.org/pdf/2306.03604> (accessed: 06.08.2025).
7. Wu Q. et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. URL: <https://arxiv.org/pdf/2308.08155> (accessed: 06.08.2025).
8. Pandya K., Holia M. Automating customer service using LangChain: Building custom open-source GPT chatbot for organizations. URL: <https://arxiv.org/pdf/2310.05421> (accessed: 06.08.2025).
9. The problem with LangChain. URL: <https://www.pelayoarbues.com/literature-notes/Articles/The-Problem-With-LangChain> (accessed: 06.08.2025).
10. Wei J. et al. Chain-of-thought prompting elicits reasoning in large language models. URL: <https://arxiv.org/pdf/2201.11903> (accessed: 06.08.2025).
11. Zeng Z. et al. MR-GSM8K: A meta-reasoning benchmark for large language model evaluation. URL: <https://arxiv.org/pdf/2312.17080> (accessed: 06.08.2025).
12. Yang Z. et al. HotpotQA: A dataset for diverse, explainable multi-hop question answering. URL: <https://arxiv.org/pdf/1809.09600> (accessed: 06.08.2025).
13. Zavras A. et al. GAIA: A global, multi-modal, multi-scale vision-language dataset for remote sensing image analysis. URL: <https://arxiv.org/pdf/2502.09598> (accessed: 06.08.2025).

Дата поступления 20.08.2025

Дата принятия 20.09.2025