УДК 004

Локальное развертывание большой языковой модели Qwen3 4B с использованием платформы Ollama

Пащенко Екатерина Александоровна
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
Целью исследования является описание и апробация последовательности установки платформы Ollama и локального запуска компактной большой языковой модели Qwen3 4B на персональном компьютере под управлением Windows. Рассматривается организация каталога хранения модели, загрузка и проверка файлов, запуск взаимодействия в терминале PowerShell и тестирование ответов на русском языке. Эксперимент подтвердил возможность автономного выполнения текстовых запросов после загрузки модели объёмом около 2,5 ГБ. На примере запроса о научной электронной библиотеке eLIBRARY.RU выявлено ограничение: при отсутствии доступа к внешним источникам модель может формировать неопределённые и недостаточно обоснованные ответы. Отмечены преимущества локального развертывания для контроля данных и его зависимость от вычислительных ресурсов компьютера.
Ключевые слова: искусственный интеллект, локальная языковая модель, Ollama, Qwen3, PowerShell, конфиденциальность.

Local deployment of the Qwen3 4B large language model using the Ollama platform

Pashchenko Ekaterina Aleksandorovna
Sholom-Aleichem Priamursky State University
Student

Abstract
The aim of this study is to describe and test a reproducible procedure for installing Ollama and running the compact Qwen3 4B large language model locally on a Windows personal computer. The paper considers model storage configuration, downloading and verification, execution in PowerShell, and Russian-language dialogue testing. The experiment confirmed that a model of approximately 2.5 GB can be downloaded and used for text generation without a remote inference service. A question concerning the eLIBRARY.RU scholarly platform illustrated a limitation of a stand-alone model: answers may be vague or unsupported without retrieval from verified external sources. Local deployment improves control over transmitted data but remains constrained by the host computer’s computational resources.
Keywords: artificial intelligence, local language model, Ollama, Qwen3, PowerShell, data privacy.

1 Введение

1.1 Актуальность

Большие языковые модели применяются для создания текстов, ответов на вопросы, обработки документов и разработки программного обеспечения. При использовании облачных сервисов обработка запросов обычно выполняется на удалённых вычислительных мощностях, что предполагает сетевое соединение и передачу пользовательских данных провайдеру. Альтернативой является локальное развертывание модели, при котором её параметры хранятся на компьютере, а генерация ответа производится непосредственно на устройстве.

Актуальность работы определяется необходимостью доступного и воспроизводимого способа ознакомления с технологией больших языковых моделей без аренды серверов и обязательной отправки запросов во внешний сервис. Особый интерес представляет сочетание компактной модели с инструментом Ollama, упрощающим загрузку и запуск нейронных сетей на Windows. При этом важно оценивать не только успешность технического запуска, но и содержательную надёжность ответов.

1.2 Обзор исследования

А. П. Москалев рассматривает ИИ-агентов на базе локальных больших языковых моделей в контексте экономической безопасности предприятия [1]. Д. М. Сенченко и Д. К. Короткова обращаются к архитектуре локального мультимодального ИИ-помощника, функционирующего в составе цифрового двойника оборудования [2]. Указанные направления свидетельствуют об интересе к обработке информации в пределах контролируемой инфраструктуры. Д. Е. Окунев и Г. А. Тимофеев посвятили работу разработке Telegram-бота для проверки новостей с применением Ollama [3]. М. А. Костин, Д. М. Давыдова и В. Е. Петров предложили RAG-архитектуру для автоматизированной проверки корпоративных документов на основе локально развернутых больших языковых моделей и интерфейса Ollama; подход предусматривает дополнение генерации поиском по документам [4]. Р. А. Долженко и А. А. Левашов рассматривают архитектуру сервиса онлайн-консультирования клиентов с интеграцией локального искусственного интеллекта [5].

1.3 Цель исследования

Цель исследования – описать и апробировать последовательность локальной установки Ollama и запуска Qwen3 4B в Windows, подтвердить работоспособность среды на примере пользовательских запросов и выявить основные преимущества и ограничения такого подхода.

2 Материалы и методы

В качестве программной среды использованы операционная система Windows, командная оболочка Windows PowerShell, платформа Ollama и языковая модель Qwen3 4B. Ollama представляет собой программную

платформу для загрузки, хранения и локального выполнения больших языковых моделей, предоставляющую консольные команды и программный интерфейс API. Для Windows предусмотрена установка приложения, работающего в фоновом режиме; локальный API по умолчанию доступен по адресу http://localhost:11434 [6].

Выбранная Qwen3 4B относится к семейству Qwen3 и содержит около четырёх миллиардов параметров. В каталоге Ollama размещён вариант размером приблизительно 2,5 ГБ [7]. Такой размер позволяет использовать модель как сравнительно доступный объект исследования, однако реальное потребление оперативной и видеопамяти в процессе выполнения больше размера файла и зависит от параметров запуска.

Метод исследования включал последовательное выполнение операций установки и загрузки, документирование результата скриншотами, диалоговое тестирование на русском языке и качественный анализ ответов. Скорость генерации, энергопотребление и количественная точность модели не измерялись, поэтому сопоставление с крупными облачными моделями носит качественный характер.

3 Результаты

На рабочем столе создан каталог local-nn, предназначенный для размещения программы и файлов нейросети. В Windows PowerShell переход в него выполняется командой:

```
cd C:\Users\Anderson\Desktop\local-nn
```

Для структурирования локального окружения предусмотрены подпапки ollama и models:

```
mkdir ollama
mkdir models
```

По умолчанию Ollama хранит данные модели в пользовательском профиле. Для размещения файлов в выбранном каталоге используется переменная окружения OLLAMA_MODELS [6]:

```
[Environment]::SetEnvironmentVariable('OLLAMA_MODELS',
   'C:\Users\Anderson\Desktop\local-nn\models', 'User')
```

Для применения значения к текущей консоли выполняется команда:

```
$env:OLLAMA_MODELS = "C:\Users\Anderson\Desktop\local-nn\models"
```

Если Ollama уже была запущена, её требуется закрыть через значок в системном трее и запустить заново, чтобы серверный процесс считал обновлённую переменную окружения.

Установщик загружается с официального сайта ollama.com/download/windows. При необходимости каталог установки самой программы задаётся параметром /DIR; данная настройка независима от расположения моделей [6]. В PowerShell можно выполнить:

```
Invoke-WebRequest -Uri "https://ollama.com/download/OllamaSetup.exe" -
OutFile "OllamaSetup.exe"
Start-Process -FilePath '.\OllamaSetup.exe' -ArgumentList
'/DIR=C:\Users\Anderson\Desktop\local-nn\ollama' -Wait
```

После установки открывается новое окно PowerShell; команда ollama --version позволяет проверить доступность программы из командной строки. Запуск фонового приложения обеспечивает обработку последующих запросов без отдельного развертывания серверной инфраструктуры.

Для получения Qwen3 4B использована команда:

ollama pull qwen3:4b

В ходе первых попыток наблюдалось зависание на стадии pulling manifest и повторные ошибки сетевых запросов к серверу хранения файлов Cloudflare R2. При последующем повторении загрузки в терминале отобразилось получение основного файла размером 2,5 ГБ, проверка контрольной суммы SHA-256, запись манифеста и сообщение success (рис. 1). Эти признаки подтверждают завершение процедуры загрузки; сам по себе вывод pulling manifest ещё не свидетельствует об успешной установке.

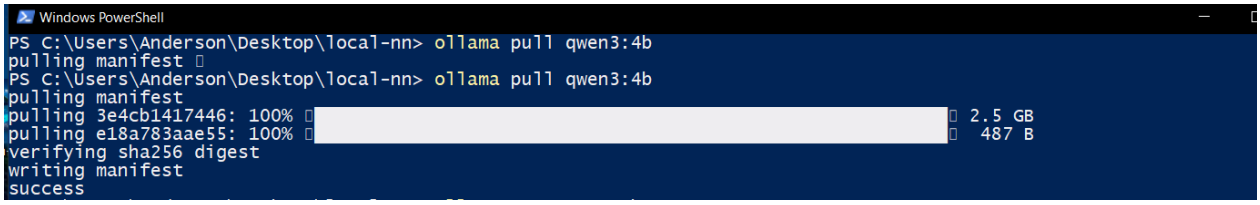


Рисунок 1. Успешная загрузка и проверка файлов Qwen3 4B

После окончания загрузки модель запускается командой:

ollama run qwen3:4b

В интерактивной консоли появляется приглашение >>>, после которого вводится пользовательский запрос. Дополнительно команды ollama list и ollama ps служат соответственно для просмотра загруженных моделей и проверки используемых вычислительных ресурсов. Завершение сеанса осуществляется командой /bye.

Первый запрос «Привет» был обработан корректно: модель сформировала русскоязычное приветствие и предложение помощи. В терминале также отображалась промежуточная область Thinking, характерная для выбранной конфигурации модели. Наличие этого текста не является признаком проверки фактов внешними источниками.

Вторым запросом стал вопрос «что такое eLIBRARY.RU». Модель предположила, что ресурс является электронной библиотекой научных материалов, но сопровождала ответ оговорками «наверное», «возможно» и рассуждениями о нескольких альтернативных значениях названия (рис. 2). В результате ответ оказался недостаточно точным и не содержал проверяемой ссылки на источник. Тест показывает конкретный случай неопределённости ответа, но не позволяет рассчитать общую частоту ошибок модели.

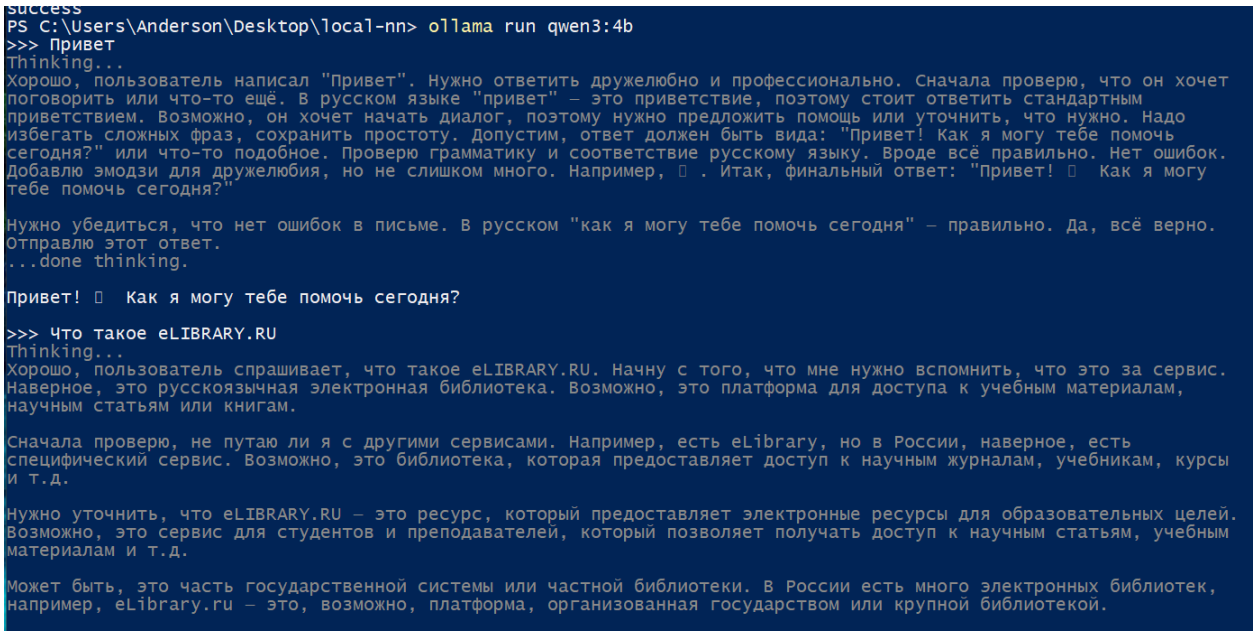


Рис. 2. Результаты диалога с локальной моделью Qwen3 4B

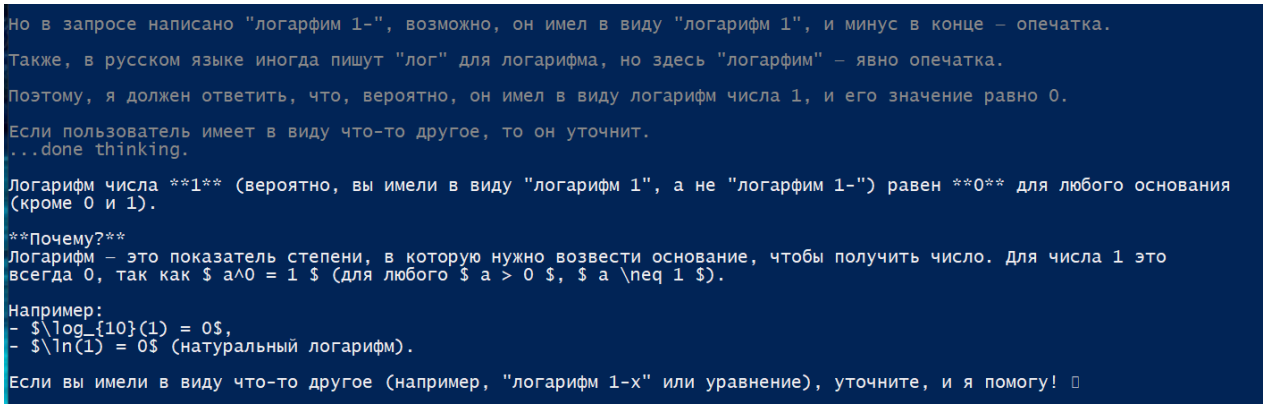


Рис. 3. Результаты диалога с локальной моделью Qwen3 4B

Последний этап – оценка условий применения. После завершения загрузки генерация может выполняться на компьютере без отправки текста запроса в облачный сервис при условии работы именно с локальной моделью и отсутствия отдельно подключённых сетевых инструментов. Это повышает контроль над конфиденциальными данными и снижает зависимость от устойчивости интернет-соединения. Вместе с тем пользователь отвечает за защиту устройства, а производительность зависит от процессора, оперативной памяти и доступности ускорения на видеокарте.

Содержательные ограничения компактной модели связаны с конечным объёмом её параметров и обучающих данных, отсутствием гарантии актуальности сведений и возможностью генерации правдоподобных, но ошибочных утверждений. Для задач, требующих точности, целесообразно дополнять локальную генерацию поиском по проверяемым документам, например, по принципу RAG [4], и сохранять человеческую проверку результата.

4 Вывод

Проведённая апробация показала техническую осуществимость локального развертывания Qwen3 4B с использованием Ollama на компьютере под управлением Windows. Выполнены подготовка каталогов, настройка хранения, загрузка модели объёмом около 2,5 ГБ, проверка целостности файлов и запуск русскоязычного диалога. Скриншоты подтверждают завершение установки модели и получение ответов на два тестовых запроса. Отдельно зафиксировано, что успешному скачиванию могут препятствовать временные сетевые сбои.

Основное преимущество подхода состоит в локальном контроле над выполнением модели и пользовательскими данными; к ограничениям относятся потребность в вычислительных ресурсах и недостаточная определённость некоторых ответов без обращения к проверенным источникам. На основании двух запросов нельзя делать количественные выводы о точности или скорости Qwen3 4B относительно облачных систем. Дальнейшее исследование может включать расширенный набор контрольных вопросов, измерение времени генерации и подключение локальной базы знаний.

Библиографический список

1. Москалев А. П. ИИ-агенты на базе локальных больших языковых моделей (LLM) для экономической безопасности предприятия // Интеллектуальная инженерная экономика и Индустрия X.0 (ИНПРОМ-2026): сборник трудов Международной научно-практической конференции: в 3 т. Санкт-Петербург, 2026. С. 86–90. URL: https://elibrary.ru/item.asp?id=91934995.
2. Сенченко Д. М., Короткова Д. К. Архитектура локального мультимодального ИИ-помощника в составе цифрового двойника технического оборудования // Перспективные исследования в современном мире: сборник статей XXIV международной научной конференции. Санкт-Петербург, 2026. С. 145–149. URL: https://elibrary.ru/item.asp?id=91996370.
3. Окунев Д. Е., Тимофеев Г. А. Разработка Телеграм-бота для проверки новостей с использованием Ollama // Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований: материалы VIII Всероссийской национальной научной конференции молодых учёных. Комсомольск-на-Амуре, 2025. С. 311–314. URL: https://elibrary.ru/item.asp?id=82811511.
4. Костин М. А., Давыдова Д. М., Петров В. Е. Реализация RAG-архитектуры для автоматизированной проверки документов в корпоративных системах хранения с использованием больших языковых моделей // Интеллектуальные технологии на транспорте. 2026. № 2 (46). С. 5–16. DOI: 10.20295/2413-2527-2026-246-5-16. URL: https://elibrary.ru/item.asp?id=91671253.
5. Долженко Р. А., Левашов А. А. Разработка архитектуры сервиса онлайн-консультирования клиентов «СпросиИИ» с интеграцией локального

искусственного интеллекта // Современные информационные технологии в образовании и научных исследованиях (СИТОНИ-2025): сборник материалов IX Всероссийской научно-технической конференции с международным участием. Донецк, 2025. С. 69–71. URL: https://elibrary.ru/item.asp?id=89141030.

6. Ollama. Windows: официальная документация. URL: https://docs.ollama.com/windows (дата обращения: 31.10.2026).
7. Ollama. Qwen3:4b: описание модели. URL: https://ollama.com/library/qwen3:4b (дата обращения: 31.10.2026).