УДК 004.81

ИССЛЕДОВАНИЕ КВАНТОВАНИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ: ОЦЕНКА ЭФФЕКТИВНОСТИ С АКЦЕНТОМ НА РУССКОЯЗЫЧНЫЕ ЗАДАЧИ

Д. Р. Пойманов[1] [0009-0001-5390-915X], М. С. Шутов[2] [0009-0009-0530-5034]

[1]*Московский государственный университет им. М. В. Ломоносова, г. Москва, Россия*

[2]*Московский физико-технический институт (национальный исследовательский университет), г. Москва, Россия*

[1]poimanovdr@my.msu.ru, [2]mihailshutov105@gmail.com

Аннотация

Квантование стало ключевой техникой сжатия и ускорения больших языковых моделей (LLM). Несмотря на то, что исследования низкобитного квантования активно развиваются применительно к англоязычным LLM, его влияние на морфологически богатые и разнородные по ресурсам языки, включая русский, остается изученным значительно хуже. Поэтому требуются дополнительные исследования этого вопроса в связи с развитием высокоэффективных русскоязычных и многоязычных LLM.

Мы провели систематическое исследование квантования предобученных моделей в эффективные 2.0—4.25 бита на параметр для современных русскоязычных LLM различного масштаба от 4 до 32 млрд параметров (4 B и 32 B). Экспериментальная часть охватывает как стандартное равномерное квантование, так и специализированные низкобитные форматы. Полученные результаты выявили несколько ключевых тенденций: i) устойчивость русскоязычных LLM к квантованию варьируется в зависимости от архитектуры и размера модели; ii) 4-битное квантование демонстрирует высокую надежность, особенно при использовании продвинутых форматов; iii) 3-битное и 2-битное квантования оказались наиболее чувствительными к указанным калибровки. Полученные эмпирические данные демонстрируют необходимость учета домена модели при использовании различных методов квантования.

***Ключевые слова**: квантование нейросетей, сжатие и оптимизация больших языковых моделей.*

ВВЕДЕНИЕ

Большие языковые модели (LLM) сегодня выступают ключевым инструментом в обработке естественного языка, обеспечивая передовые результаты в таких задачах, как ответы на вопросы [1], ведение диалога [2], генерация кода [3] и рассуждения [4–6]. Однако стремительное увеличение размера моделей — до десятков и сотен миллиардов параметров — сопровождается резким ростом потребностей в вычислительных ресурсах и памяти, что делает проблему эффективного развертывания одной из центральных для современной исследовательской повестки. Одним из наиболее эффективных подходов к решению проблемы является квантование, при котором веса моделей преобразуются в более компактные представления. Такой подход позволяет существенно ускорить процесс инференса и снизить затраты памяти [7–10]. Недавние разработки методов квантования, включая GPTQ, AWQ, SmoothQuant и QTIP, показали, что при использовании специализированных алгоритмов квантования возможно сохранить высокую производительность даже в условиях агрессивного сжатия [9–11].

Несмотря на значительные достижения в области квантования нейросетей, большинство исследований сосредоточено на англоязычных или многоязычных LLM, что оставляет существенный пробел в изучении моделей для русского языка. В последние годы российское NLP-сообщество представило ряд крупномасштабных открытых моделей, включая T-Pro 2.0 [12], YaGPT [13] и RuAdaptQwen [14]. Эти разработки расширяют и адаптируют возможности многоязычных базовых архитектур, таких как Qwen [15], к задачам, ориентированным на русский язык. Хотя названные модели демонстрируют конкурентоспособные результаты на ряде бенчмарков, они, как правило, распространяются в форматах полной точности (FP16 или BF16), а доступные версии с квантованием, если они вообще присутствуют, нередко сопровождаются заметной деградацией качества. Это ограничивает практическое применение русскоязычных LLM в условиях дефицита ресурсов — например, на мобильных устройствах или в промышленных системах с требованиями к низкой задержке по памяти.

Особый интерес и актуальность исследования подкрепляются тем, что ведущие исследовательские группы все чаще выпускают модели в квантованных форматах по умолчанию. Так, компания OpenAI представила оптимизированные 4-битовые версии своих моделей [16], а в DeepSeek показано [17], что и обучение, и инференс модели непосредственно с использованием 8-битной арифметики позволяют достигать сопоставимой производительности при существенном росте эффективности. Эти тенденции свидетельствуют о том, что в будущем практическое внедрение LLM может опираться не столько на модели с высокой числовой точностью, сколько на тщательно спроектированные низкобитные представления. В то же время для русскоязычных LLM данное направление остается недостаточно изученным: отсутствует систематическая оценка того, как различные методы квантования и разрядность влияют на качество моделей при решении ключевых лингвистических и логических задач.

В настоящей работе мы стремимся восполнить этот пробел, проводя всестороннее исследование влияния квантования на LLM, адаптированные к русскому языку. Особое внимание уделено моделям, охватывающим различные масштабы и архитектуры, а именно на Qwen3-4B, RuAdaptQwen3-4B, Qwen3-32B, RuAdaptQwen3-32B и T-Pro 2.0-32B. Для каждой из названных моделей мы провели серию экспериментов по квантованию: i) скалярное равномерное квантование после обучения (Post Training Quantization, PTQ) в 4, 3 и 2 бита на параметр; ii) векторное квантование в эффективные 2 бита на вес методом QTIP и iii) квантование весов модели в специализированные форматы MXFP4 и MXINT4. Оценка деградации модели проведена на бенчмарках, включающих общеязыковые (PIQA, MMLU) и русскоязычные (PIQA-RU, MMLU-RU) наборы данных, что позволило выявить взаимосвязь между стратегией квантования, битностью и языковой адаптацией.

Статья состоит из двух основных частей.

1. В первой части представлено первое систематическое исследование квантования для LLM, адаптированных для русского языка, в котором освещены актуальные проблемы, а также проведено сравнение с моделями, ориентированными на английский язык.

2. Вторая часть посвящена детальному сравнительному анализу компромиссов между различными подходами к квантованию, включая скалярное квантование с калибровкой, агрессивное векторное квантование и наивное квантование в специализированные форматы. При этом было учтено влияние двух типов калибровочных данных — англоязычного корпуса RedPajama и русскоязычного T-Wix.

Мы считаем, что настоящее исследование формирует методологическую основу для последующей разработки эффективных методов адаптации и внедрения специализированных больших языковых моделей с учетом различных лингвистических особенностей.

ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ

Современные методы квантования можно разделить на две группы в зависимости от стратегии сжатия (PTQ и QAT) и методологии сжатия (скалярное и векторное квантование). PTQ (Post-Training Quantization, квантование после обучения) относится к подходам, при которых модель сжимается после того, как она уже прошла обучение. Методы PTQ являются высокоэффективными с точки зрения вычислительных затрат, поскольку дорогостоящий этап обучения уже завершен. QAT (Quantization-Aware Training, Обучение с учетом квантования) — это обучение, как правило, всех параметров модели с имитацией эффектов квантования. Хотя подходы QAT обычно обеспечивают более высокую эффективность сжатия, они требуют значительно больше вычислительных ресурсов даже в сравнении с обучением без квантования.

В настоящем исследовании мы делаем акцент на PTQ-методах. Рассмотрим сначала ключевые принципы PTQ, включая адаптивное округление (adaptive rounding), поблочную оптимизацию (block-wise fine-tuning), эффективное дообучение модели (parameter-efficient end-to-end fine-tuning or PEFT), а также алгоритмы скалярного и векторного квантования.

Адаптивное округление (*Adaptive Rounding*) представляет собой метод посттрейнинговой квантизации, при котором вместо стандартного округления к ближайшему значению используется итеративный алгоритм, поэтапно кванти-

зирующий подмножество весов слоя LLM. На каждом шаге алгоритм минимизирует ошибки, возникающие в выходных активациях слоя вследствие квантизации на предыдущих итерациях. Наиболее известные подходы к квантизации LLM с использованием адаптивного округления — GPTQ [8] и LDLQ [18] — рассматривают выход каждого линейного слоя в качестве локальной целевой функции при минимизации ошибки квантизации, что позволяет проводить процедуру квантизации модели параллельно и без большой калибрационной выборки.

Блочный PTQ [19, 20] представляет собой продвинутую технику дистилляции знаний, при которой блоки квантованной модели (студента) обучаются на основе соответствующих блоков оригинальной модели (учителя) с использованием функции потерь, наложенной на активации. Такой подход значительно сокращает вычислительный граф в процессе оптимизации и, как следствие, снижает затраты ресурсов.

Эффективное дообучение. После этапа блочного PTQ используют метод PEFT [21, 22] для восстановления качества квантованной модели и оптимизации ее производительности. В отличие от традиционного обучения, которое корректирует все параметры сети, PEFT фокусируется только на небольшом поднаборе важных параметров, которые в наибольшей степени влияют на точность вывода модели.

В контексте квантования LLM наибольший интерес представляет квантование весов линейных слоев, поскольку эти слои содержат большую часть параметров модели (обычно >95% весов модели). Квантование линейных слоев позволяет существенно снизить как вычислительные затраты на умножение матриц, так и объем занимаемой памяти. В настоящее время наиболее широко используются методы скалярного квантования, однако в условиях низкобитного сжатия (2 бита на вес или меньше) методы векторного квантования позволяют получать меньшую просадку в точности.

Скалярное квантование. При скалярном квантовании (Scalar quantization) каждое вещественное значение параметра модели заменяется на значение из дискретного множества уровней квантования. Таким образом, непрерывное пространство параметров аппроксимируется конечным набором возможных значений, что позволяет значительно сократить объем хранимых данных. В контексте

сжатия нейронных сетей задача квантования заключается в минимизации искажения, возникающего при замене исходных весов их квантованными аналогами. В простейшем случае минимизируется ошибка, определяемая как разность между исходным и квантованными параметрами, однако при низкобитном сжатии становится необходим учет изменения отклика модели на калибровочном наборе данных, чтобы сохранить качество предсказаний.

В процессе скалярного квантования каждому параметру сопоставляется индекс соответствующего уровня квантования, который сохраняется в низкобитном формате. Например, при 4-битном квантовании два параметра могут быть закодированы в одном байте памяти. Это обеспечивает компактное хранение параметров и возможность восстановления приближенного вектора весов модели при необходимости.

Векторное квантование. В отличие от скалярного квантования, векторное квантование (*Vector Quantization, VQ*) предполагает квантование целых векторов весов нейронной сети, а не отдельных скалярных параметров. Каждая группа весов квантуемого слоя (вектор весов) заменяется одним из векторов из заранее определенного набора — кодовой книги (*codebook*). В результате квантования каждому вектору весов сопоставляется индекс выбранного вектора кодовой книги, что позволяет эффективно хранить параметры в сжатом виде. В результате достигается значительное уменьшение объема модели при умеренной потере точности. В настоящее время ведущими методами низкобитного квантования LLM (эффективные 2 бита на вес и ниже) с точки зрения компромисса между сжатием и качеством модели являются именно методы векторного квантования — AQLM [23], Quip# [24] и QTIP [25].

В нашем экспериментальном исследовании мы использовали методы как скалярного, так и векторного квантования. Мы реализовали методологии, представленные в оригинальных работах, однако скорректировали некоторые гиперпараметры и наборы калибрационных датасетов для соответствия целям исследования.

ЭКСПЕРИМЕНТЫ

Нами были использованы открытые языковые модели и наборы данных на русском и английском языках. Для анализа влияния различных методов квантования мы рассмотрели как подходы, основанные на калибровке (EfficientQAT, QTIP), так и методы без калибровки (квантование весов в форматы Microscaling). В качестве оценки деградации моделей после квантования были использованы открытые англоязычные и русскоязычные бенчмарки.

Модели

Оценим разнообразный набор современных больших языковых моделей, включая как мультиязычные, так и варианты, адаптированные для русского языка. В частности, рассмотрим модель Qwen3-4B [26] и ее адаптированную для русского языка версию RuAdaptQwen3-4B [27], представляющие собой компактные модели на основе архитектуры трансформер для решения задач обработки естественного языка. В качестве больших моделей исследуем Qwen3-32B [26] и T-Pro-2.0-32 [21], причем последняя является одной из самых мощных открытых LLM, ориентированных на русский язык и выпущенных на данный момент. Такой выбор позволяет нам сравнить эффекты квантования для моделей различных размеров (4 B и 32 B) и для разных языковых доменов (многоязычные и оптимизированные для русского языка модели).

Методы квантования нейросетей

В экспериментах по квантованию больших языковых моделей мы применили несколько методов, которые продемонстрировали свою эффективность как в академических исследованиях, так и в промышленных приложениях, в частности, скалярное квантование на основе Learning Step Quantization (LSQ) в рамках EfficientQAT [19], преобразование весов модели в числовой формат Microscaling [28] и QTIP [25], современный метод векторного квантования, который использует trellis сжатие для максимально эффективного распределения битов при векторном квантовании. Краткие описания каждого метода даны ниже (см. рис. 1–3).

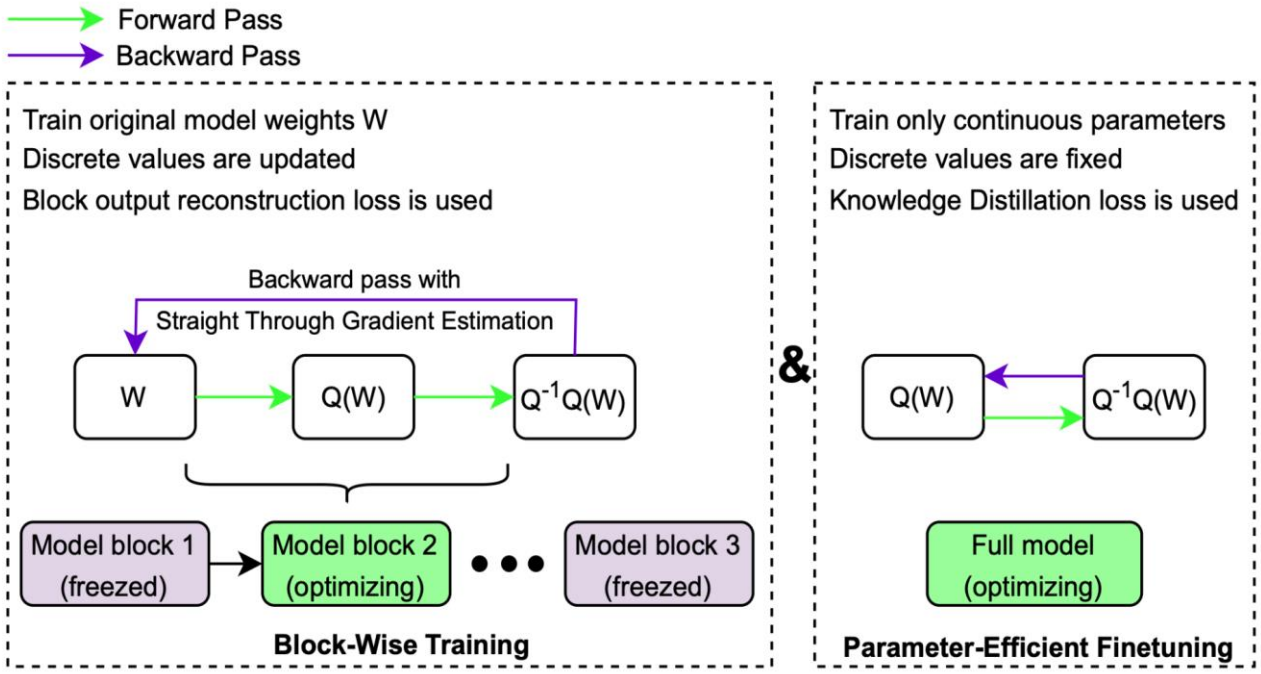

Рис. 1. Пайплайн квантования EfficientQAT [19]. Посттрейнинговое квантование предполагает 2 этапа: 1) поблочная дистилляция знаний и 2) эффективное по затрачиваемым ресурсам дообучение модели на выходы оригинальной модели.

EfficientQAT — это метод обучения, изначально разработанный для скалярного квантования и основанный на двухэтапной схеме: i) блочная дистилляция знаний из оригинальной модели (учителя) в квантованную (ученик) и ii) дообучение ограниченного числа параметров (PEFT) квантованной модели. Такой подход представляет собой ресурсоэффективный вариант постобучающего квантования, позволяющий сохранять качество практически без потерь даже при агрессивном квантовании, требуемом при жестких вычислительных ограничениях.

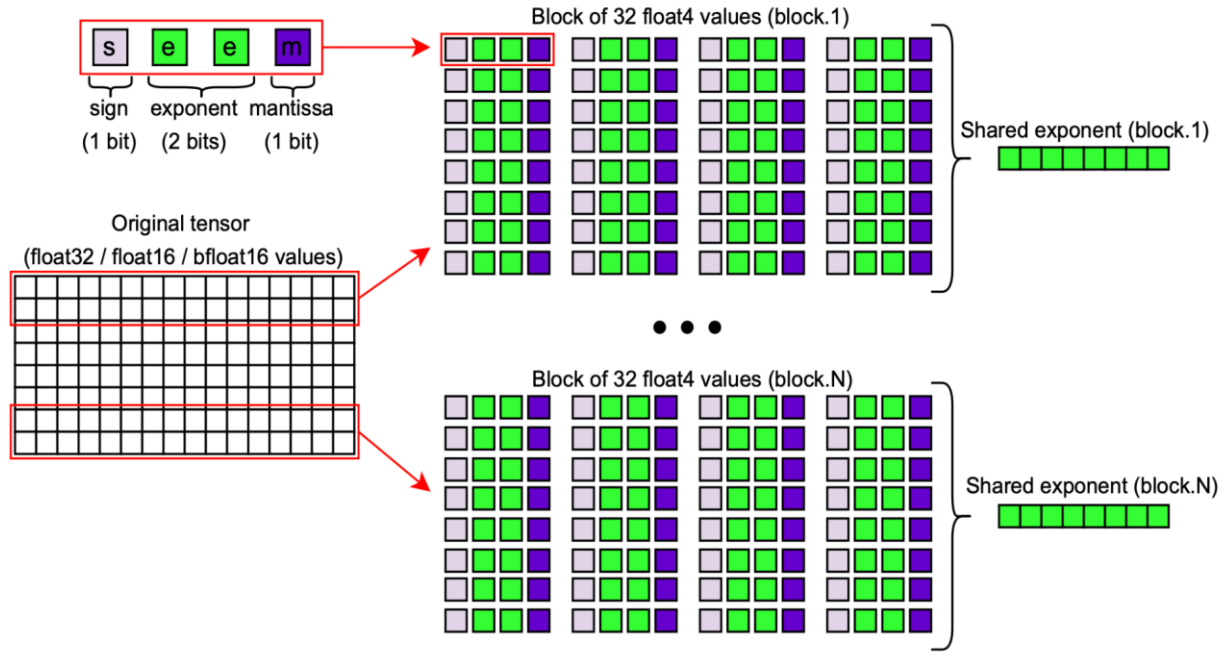

Рис. 2. Визуализация квантования тензора bfloat16 значений в MXFP4 формат [28]. Каждый блок из 32 оригинальных весов представляется в виде 32 float4 чисел и одной общей экспоненты.

Microscaling формат данных. Формат MX (Microscaling) представляет собой специализированное числовое представление, разработанное для снижения объемов памяти и ускорения инференса. Матрица в таком формате разделена на группы элементов, каждый из которых хранится в низкоразрядном формате (например, FP4 или INT4), при этом каждой группе присваивается коэффициент масштабирования, являющийся общей экспонентой для всех элементов группы. Формат MX нативно поддерживается в последних поколениях графических процессоров NVIDIA (например, в архитектуре Blackwell), что упрощает как процесс сжатия, так и аппаратное выполнение вычислений.

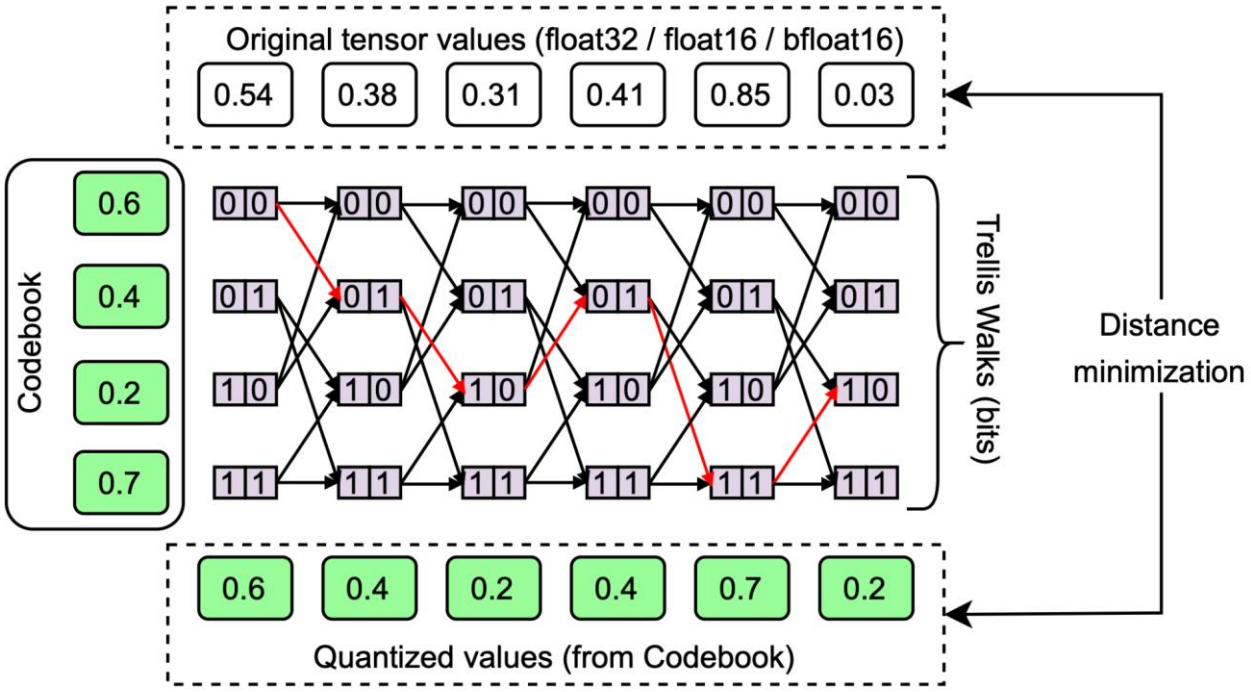

Рис. 3. Визуализация получения квантованных весов методом векторного квантования QTIP [25]. Индекс вектора формируется так, что биты соседних скаляров в векторе переиспользуются. Для получения оптимальной конфигурации используется алгоритм Витерби (Viterbi).

QTIP представляет собой современный метод векторного квантования, в котором используются умножения весов и активаций LLM на рандомизированные матрицы Адамара для снижения числа выбросов, а затем применяется высокоэффективное векторное квантование преобразованных весов с использованием техники адаптивного округления BlockLDLQ. Такой подход позволяет сжимать LLM до эффективных 2 бит на вес при сохранении качества генерации текстов сравнимым с оригинальной моделью. На данный момент QTIP можно рассматривать как одну из наиболее передовых технологий векторного PTQ для крупномасштабных LLM.

Во всех наших экспериментах квантованию подвергались только веса линейных слоев трансформер-блоков в соответствии с наиболее распространенными практиками квантования больших языковых моделей. Мы изучили как скалярное квантование (равномерное квантование в 2, 3 и 4 бита, Microscaling-фор-

маты), так и низкобитное векторное квантование (QTIP). В случае скалярного квантования в 3 и 4 бита мы применили групповое масштабирование и смещение весов с размером группы 128 параметров, что дало эффективные 3.25 и 4.25 бит на вес соответственно. При квантовании в скалярные 2 бита мы использовали размер группы в 64 элемента (эффективные 2.5 бит на вес), а также стандартные Microscaling форматы (MXFP4 и MXINT4) с группами по 32 элемента с компактными масштабами (эффективные 4.25 бит на вес). Векторное квантование QTIP реализует наиболее агрессивное сжатие — до эффективных 2.0 бит на вес.

Данные для обучения

Были использованы два набора данных для калибровки. RedPajama [29] — это открытый корпус для предварительного обучения, созданный с целью воспроизведения и расширения исходного набора веб-источников и систематизации знаний (CommonCrawl, C4, GitHub, Wikipedia, Books, arXiv, StackExchange). Первоначально он был выпущен как «чистый» набор данных объемом 1,2 ТБ (V1), а позднее расширен до версии RedPajama-V2, включающей 30 ТБ отфильтрованных токенов (более 100 ТБ в необработанном виде) на пяти языках и дополненной метаданными качества для отбора. В наших экспериментах в качестве калибровочного корпуса мы использовали 8 млн токенов RedPajama (~40 Mb текста), преимущественно англоязычных, отражающих широкий спектр веб-ресурсов и источников знаний.

Другой набор, T-Wix [30], представляет собой российский набор данных для дообучения нейросетей, ориентированный на выполнение инструкций и задач по рассуждению на определенные темы. Корпус включает два раздела: «Общий» (468 тыс. примеров, охватывающих математику, естественные науки, программирование, общие знания, ролевые сценарии и др.) и «Рассуждения» (31 тыс. примеров с подробными пошаговыми трассировками решений). В наших экспериментах T-Wix был использован для калибровки моделей в соответствии с русскоязычным распределением данных и форматами инструкций, что позволило сопоставимо оценить влияние калибровки при различных режимах квантования.

Бенчмарки

Для оценки мы применили комбинацию тестов, а именно: вычислили перплексию квантованных языковых моделей и их точность ответов на вопросы различной сложности. Для оценки качества языкового моделирования были задействованы корпус WikiText [31] и его русский аналог WikiText-Ru [32], которые обеспечивают стандартизированную оценку перплексии модели на естественном тексте. Для оценки эффективности рассуждений и ответов на вопросы были использованы MMLU [33] и PIQA [34], два широко используемых английских бенчмарка, охватывающих многозадачные знания и рассуждение на общие темы. Кроме того, мы включили их русские адаптации — MMLU-Ru [35] и PIQA-Ru [36] — для специальной проверки устойчивости моделей на русском языке. Для запуска тестов была применена стандартизированная библиотека lm-evaluation-harness [37]. В совокупности этот набор тестов позволил нам измерить как общую перплексию, так и зависимость точности выполнения конкретных задач в зависимости от языка.

РЕЗУЛЬТАТЫ

Представим результаты для четырех моделей Qwen3-4B, RuAdaptQwen3-4B, Qwen3-32B и T-Pro 2.0-32B в четырех режимах квантования: скалярное квантование методом LSQ от 2 до 4 бит и векторное квантование QTIP в режиме 2 бита на вес. Для каждого режима калибровка выполнялась с использованием двух текстовых корпусов (RedPajama и T-Wix) по 8 M токенов. Показатели качества включают в себя метрику качества языковых моделей – перплексию – на датасете WikiText (с предложениями на английском языке) и WikiText-RU (с предложениями, переведенными на русский язык), а также долю правильных ответов на выбранных QA (Question Answering) бенчмарках MMLU, MMLU-Ru, PIQA и PIQA-Ru для тестирования качества. Сначала обсудим перплексию, потом точность ответов на вопросы, а затем проанализируем зависимость качества моделей от калибровочных данных, отдельно уделив внимание влиянию векторного и скалярного квантования при 2 битах. Полные результаты представлены в табл. 1–3.

Во всех четырех моделях скалярное квантование до 4 бит на вес в значительной степени сохраняло качество работы моделей (метрика близка к модели

в оригинальном BF16-формате) как на корпусе WikiText, так и на WikiText-RU. В частности, модель T-Pro-IT-2.0-32B демонстрирует наименьший рост перплексии, оставаясь близкой к полной точности на обоих корпусах, в то время как модели объемом 4 B оказались несколько более чувствительными, особенно на корпусе WikiText-RU. Модель Qwen3-32B проявила такую же устойчивость, характерную для крупных моделей: ее перплексия в 4-битном режиме аномально снижается на русскоязычных данных. В целом квантование до 4 битов является надежным вариантом для развертывания всех исследуемых моделей, при этом относительное отклонение в качестве для вариантов объемом 4 B больше на корпусе WikiText-RU, чем на WikiText.

Снижение битности до 3 бит приводит к значительному росту перплексии на обоих корпусах у всех моделей. Снижение качества более выражено на корпусе WikiText-RU, что указывает на повышенную чувствительность к квантованию для русского языка. Большие модели (Qwen3-32B, T-Pro-IT-2.0-32B) сохраняют бóльшую стабильность по сравнению с Qwen3-4B, однако разрыв в качестве между английским и русским языками увеличивается по сравнению с 4-битным режимом. Это позволяет предположить, что морфологическая сложность русского языка и различия в токенизации усиливают шум, вызванный агрессивным квантованием.

Табл. 1. Результаты оценки квантованных моделей с числом параметров 4 B

Точность	Данные	W2	W2-Ru	MMLU	MMLU-Ru	PIQA	PIQA-Ru
Qwen3-4B							
FP16	–	11.7	6.94	70.0	62.1	76.3	64.3
scalar 4 bit	RedPaj.	12.3	7.34	68.5	61.0	75.7	63.4
	T-Wix	12.5	7.27	68.8	61.3	75.8	63.7
scalar 3 bit	RedPaj.	15.9	9.10	63.2	54.2	74.5	60.7
	T-Wix	15.9	8.78	61.6	54.2	74.4	62.7

scalar 2 bit	RedPaj.	14.8	12.0	48.6	38.6	70.8	56.7
	T-Wix	18.5	8.56	47.2	41.6	70.2	60.4
QTIP 2 bit	RedPaj.	13.5	11.5	52.7	37.7	73.1	57.5
	T-Wix	19.9	8.21	49.6	42.2	72.9	61.8
RuadaptQwen3-4B							
FP16	–	9.09	11.0	68.9	62.6	77.5	67.7
scalar 4 bit	RedPaj.	9.44	11.6	67.4	60.9	77.4	66.6
	T-Wix	9.49	11.6	68.0	60.7	77.3	67.3
scalar 3 bit	RedPaj.	11.2	14.5	61.6	54.2	75.4	64.3
	T-Wix	11.4	14.0	59.9	52.2	75.9	64.5
scalar 2 bit	RedPaj.	13.5	24.8	48.3	37.3	71.9	59.3
	T-Wix	16.4	18.9	45.2	39.6	70.3	61.9
QTIP 2 bit	RedPaj.	12.4	19.8	49.0	38.3	67.9	53.1
	T-Wix	14.6	17.9	45.7	41.4	69.6	55.4

Табл. 2. Результаты оценки квантованных моделей с числом параметров 32 B

Точность	Данные	W2	W2-Ru	MMLU	MMLU-Ru	PIQA	PIQA-Ru
Qwen3-32B							
FP16	–	6.67	4.44	81.9	76.6	81.9	70.5
scalar 4 bit	RedPaj.	6.80	4.38	81.2	76.0	81.7	70.0
	T-Wix	6.82	4.37	81.0	76.1	81.9	70.4
scalar 3 bit	RedPaj.	7.59	4.88	80.3	74.4	80.9	69.2
	T-Wix	7.65	4.78	80.0	74.5	81.7	70.4
scalar 2 bit	RedPaj.	9.88	6.04	72.1	64.2	77.3	65.1
	T-Wix	9.24	5.24	72.0	67.2	78.1	67.1
QTIP 2 bit	RedPaj.	9.32	12.4	71.2	65.1	77.2	66.7
	T-Wix	13.0	7.99	70.4	66.5	77.9	67.1
T-pro-it-2.0							
FP16	–	5.53	6.96	83.6	78.7	81.9	71.2
scalar 4 bit	RedPaj.	5.67	7.18	83.1	77.1	81.8	71.0
	T-Wix	5.66	7.21	82.2	77.9	81.9	71.0
scalar 3 bit	RedPaj.	6.20	8.19	81.3	75.1	81.6	69.9
	T-Wix	6.30	8.03	80.3	75.2	81.8	70.7
scalar 2 bit	RedPaj.	7.53	16.6	73.7	63.9	78.3	65.8
	T-Wix	8.56	11.4	73.8	68.5	78.0	68.1

QTIP 2 bit	RedPaj.	7.77	34.3	72.6	65.0	78.2	66.9
	T-Wix	10.4	13.9	71.5	66.2	78.9	67.4

При использовании скалярного 2-битного квантования наблюдается существенный рост перплексии, особенно на корпусе WikiText-RU. Наиболее заметное увеличение демонстрируют модели с 4 B параметров, в то время как 32 B-модели остаются относительно более стабильными, хотя их качество также заметно снижается по сравнению с 3-битным режимом. На практике 2-битное скалярное квантование оказывается чрезмерно агрессивным для сохранения качества генерации текстов на русском языке и требует применения методов дообучения.

На QA бенчмарках 4-битное скалярное квантование сохраняет точность, незначительно уступая несжатым моделям. Для моделей Qwen3-32B и T-Pro-IT-2.0-32B высокие показатели на MMLU сохраняются, а на MMLU-RU наблюдается лишь минимальный регресс. Аналогичная картина наблюдается для бенчмарков PIQA и PIQA-RU. Модели с 4 B параметрами демонстрируют несколько большее снижение точности, причем их русскоязычные варианты (MMLU-RU, PIQA-RU) страдают сильнее, чем англоязычные. Тем не менее это снижение остается в пределах, приемлемых для большинства практических применений.

При 3-битной точности все четыре показателя точности снижаются во всех моделях. Снижение незначительно для моделей 32B и более заметно для Qwen3-4B. Показатели на русскоязычных тестах (MMLU-RU, PIQA-RU) падают больше, чем на англоязычных, что соответствует наблюдениям, сделанным ранее по перплексии, и подчеркивает, что ведение рассуждений и анализ текстов моделями в русском языке более чувствительны к шуму квантования.

Табл. 3. Результаты оценки квантованных моделей с числом параметров 32 B (scalar vs microscaling)

Точность	W2	W2-Ru	W2	W2-Ru
	Qwen3-32B		T-pro-it-2.0	
FP16	6.67	4.44	5.53	6.96
Scalar 4bit	6.80	4.38	5.67	7.18
	6.82	4.37	5.66	7.21
MXFP4	11.5	10.4	11.7	8.36
MXINT4	12.7	8.55	5.83	7.51

При скалярном 2-битном квантовании снижение точности на QA становится существенным, особенно на бенчмарках MMLU-RU и PIQA-RU. Для моделей объемом 4 B данный режим квантования, как правило, не соответствует приемлемым порогам качества для промышленного развертывания без дополнительной адаптации. Хотя 32 B-модели по-прежнему превосходят 4 B-модели при той же разрядности, наблюдаемое снижение качества по сравнению с 3-битным режимом подтверждает, что прямое 2-битное квантование является рискованным для QA задач.

Сравнение скалярного и векторного квантования при 2 битах. Во всех четырех моделях скалярное 2-битное квантование демонстрирует незначительное превосходство над векторным как по перплексии, так и по точности на QA бенчмарках. Однако векторное квантование обеспечивает более низкую среднюю эффективную разрядность, что приводит к дополнительной экономии памяти (эффективные 2.5 бит в случае скалярного квантования с группами по 64 элемента против эффективных 2 бит QTIP). Компромисс между методами особенно заметен на русскоязычных бенчмарках (WikiText-RU, MMLU-RU, PIQA-RU), где скалярное 2-битное квантование сохраняет бóльшую стабильность, в особенности

для моделей с 4 B параметрами. В случае более крупных 32 B-моделей разрыв между методами сокращается, и векторное квантование, откалиброванное с помощью русскоязычного корпуса T-Wix, в некоторых случаях приближается по качеству к скалярному. Таким образом, на практике скалярное 2-битное квантование остается предпочтительным, когда приоритетной задачей является сохранение качества модели, тогда как векторное квантование становится привлекательной альтернативой в случаях, требующих максимальной эффективности использования памяти.

Корреляции и характер ошибок. Мы наблюдаем положительную корреляцию между ростом перплексии и снижением точности на QA бенчмарках, причем на русскоязычных задачах эта связь выражена сильнее. Анализ ошибок на бенчмарках MMLU-RU и PIQA-RU показал, что низкобитное скалярное квантование увеличивает количество ошибок, связанных с согласованием (например, падежным, родовым), пониманием устойчивых выражений, а также с обработкой слов с длинными формами. Эта картина является закономерной и соответствует лингвистическим особенностям русского языка.

ЗАКЛЮЧЕНИЕ

Проведенное исследование дает первую систематическую оценку квантования больших языковых моделей, адаптированных под русский язык. Наши результаты показали, что хотя 4-битное квантование остается высоконадежным даже для морфологически сложного русского языка, переход на 3 бита вызывает значительное снижение устойчивости в качестве при генерации текстов, особенно при сжатии небольших моделей и тестирования их на специализированных русскоязычных бенчмарках. При 2-битном квантовании наивные подходы оказываются практически неприменимыми для русскоязычных LLM, однако калибровки на русскоязычных корпусах данных частично восстанавливают производительность. В целом оптимизированные для русского языка модели, такие как T-Pro-2.0-32B, демонстрируют более высокую устойчивость к квантованию по сравнению с мультиязычными аналогами, что подчеркивает важность как масштаба модели, так и языковой адаптации. Эти результаты свидетельствуют о том,

что успешное развертывание сжатых русскоязычных LLM требует не только технических инноваций в области квантования, но и тщательного учета языково-специфичной калибровки и оценки.

СПИСОК ЛИТЕРАТУРЫ

1. *Shavrina T. et al.* RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020. P. 4717–4726. https://doi.org/10.18653/v1/2020.emnlp-main.381

2. *Mendonça J., Lavie A., Trancoso I.* On the Benchmarking of LLMs for Open-Domain Dialogue Evaluation // Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024). 2024. P. 1–12. https://doi.org/10.48550/arXiv.2407.03841

3. *Liu J. et al.* Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation //Advances in Neural Information Processing Systems. 2023. Vol. 36. P. 21558–21572. https://doi.org/10.48550/arXiv.2305.01210

4. *Hendrycks D. et al.* Measuring massive multitask language understanding, 2021 // International Conference on Learning Representations. 2021. https://doi.org/10.48550/arXiv.2009.03300

5. *Clark P. et al.* Think you have solved question answering? try arc, the ai2 reasoning challenge // arXiv preprint arXiv:1803.05457. 2018. https://doi.org/10.48550/arXiv.1803.05457

6. *Zellers R. et al.* HellaSwag: Can a Machine Really Finish Your Sentence? // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. P. 4791–4800. https://doi.org/10.48550/arXiv.1905.07830

7. *Dettmers T. et al.* Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale // Advances in neural information processing systems. 2022. Vol. 35, P. 30318–30332. https://doi.org/10.48550/arXiv.2208.07339

8. *Frantar E. et al.* OPTQ: Accurate post-training quantization for generative pre-trained transformers // 11th International Conference on Learning Representations. 2023. https://doi.org/10.48550/arXiv.2210.17323

9. *Lin J. et al.* Awq: Activation-aware weight quantization for on-device llm compression and acceleration // Proceedings of machine learning and systems. 2024. Vol. 6. P. 87–100. https://doi.org/10.1145/3714983.3714987

10. *Xiao G. et al.* Smoothquant: Accurate and efficient post-training quantization for large language models // International conference on machine learning. PMLR, 2023. P. 38087 –38099. https://doi.org/10.48550/arXiv.2211.10438

11. *Tseng A. et al.* Qtip: Quantization with trellises and incoherence processing // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 59597–59620. https://doi.org/10.48550/arXiv.2406.11235

12. *T-Tech.* T-pro-2.0. – Hybrid reasoning model based on Qwen3-32B // HuggingFace.co: The collaboration platform. 2025. URL: https://huggingface.co/t-tech/T-pro-it-2.0

13. *Yandex company.* YandexGPT // HuggingFace.co: The collaboration platform. 2025. URL: https://huggingface.co/yandex/YandexGPT-5-Lite-8B-instruct

14. *Tikhomirov M., Chernyshev D.* Facilitating large language model russian adaptation with learned embedding propagation // Journal of Language and Education. 2024. Vol. 10. No. 4 (40). P. 130–145. https://doi.org/10.48550/arXiv.2412.21140

15. *Team Q. et al.* Qwen2 technical report // arXiv preprint arXiv:2407.10671. 2024. Vol. 2. P. 3. https://doi.org/10.48550/arXiv.2407.10671

16. *Agarwal S. et al.* gpt-oss-120b & gpt-oss-20b Model Card // arXiv e-prints. 2025. P. arXiv: 2508.10925. https://doi.org/10.48550/arXiv.2508.10925

17. *Liu A. et al.* DeepSeek-V3 Technical Report // arXiv e-prints. 2024. P. arXiv: 2412.19437. https://doi.org/10.48550/arXiv.2412.19437

18. *Chee J. et al.* Quip: 2-bit quantization of large language models with guarantees // Advances in Neural Information Processing Systems. 2023. Vol. 36, P. 4396 – 4429. https://doi.org/10.48550/arXiv.2307.13304

19. *Chen M. et al.* Efficientqat: Efficient quantization-aware training for large language models // Annual Meeting of the Association for Computational Linguistics. 2025. Vol. 1. P. 10081–10100. https://doi.org/10.48550/arXiv.2407.11062

20. *Shao W. et al.* OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models // The Twelfth International Conference on Learning Representations. 2024. https://doi.org/10.48550/arXiv.2308.13137

21. *Hu E. J. et al.* Lora: Low-rank adaptation of large language models // International Conference on Machine Learning. 2022. Vol. 1, No. 2. P. 3. https://doi.org/10.48550/arXiv.2106.09685

22. *Han Z. et al.* Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey // arXiv e-prints. 2024. P. arXiv: 2403.14608. https://doi.org/10.48550/arXiv.2403.14608

23. *Egiazarian V. et al.* Extreme compression of large language models via additive quantization // Proceedings of the 41st International Conference on Machine Learning. 2024. P. 12284–12303. https://doi.org/10.48550/arXiv.2401.06118

24. *Tseng A. et al.* QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks // International Conference on Machine Learning. PMLR, 2024. P. 48630–48656. https://doi.org/10.48550/arXiv.2402.04396

25. *Tseng A. et al.* Qtip: Quantization with trellises and incoherence processing // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 59597–59620. https://doi.org/10.48550/arXiv.2406.11235

26. *Yang A. et al.* Qwen3 technical report // arXiv e-prints. 2025. P. arXiv: 2505.09388. https://doi.org/10.48550/arXiv.2505.09388

27. *Achiam J. et al.* GPT-4 Technical Report // arXiv e-prints. 2023. arXiv: 2303.08774. https://doi.org/10.48550/arXiv.2303.08774

28. *Darvish Rouhani B. et al.* Microscaling data formats for deep learning // arXiv e-prints. 2023. P. arXiv: 2310.10537. https://doi.org/10.48550/arXiv.2310.10537

29. *Weber M. et al.* Redpajama: an open dataset for training large language models // Advances in neural information processing systems. 2024. Vol. 37. P. 116462–116492. https://doi.org/10.52202/079017-3697

30. *Potapov A.* T-Wix – Russian supervised fine-tuning (SFT) dataset // HuggingFace.co: The collaboration platform. 2025. URL: https://huggingface.co/datasets/t-tech/T-Wix

31. *Merity S. et al.* Pointer Sentinel Mixture Models // International Conference on Learning Representations. 2017. https://doi.org/10.48550/arXiv.1609.07843

32. *Korablinov V., Braslavski P.* RuBQ: A Russian dataset for question answering over Wikidata // International Semantic Web Conference. Cham: Springer International Publishing. 2020. P. 97–110. https://doi.org/10.1007/978-3-030-62466-8_7

33. *Li H. et al.* CMMLU: Measuring massive multitask language understanding in Chinese // Findings of the Association for Computational Linguistics. 2024. P. 11260–11285. https://doi.org/10.48550/arXiv.2306.09212

34. *Bisk Y. et al.* Piqa: Reasoning about physical commonsense in natural language // Proceedings of the AAAI conference on artificial intelligence. 2020. Vol. 34. №. 05. P. 7432–7439. https://doi.org/10.1609/aaai.v34i05.6239

35. *Fenogenova A. et al.* MERA: A Comprehensive LLM Evaluation in Russian //Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024. P. 9920–9948. https://doi.org/10.18653/v1/2024.acl-long.534

36. *Chirkin A. et al.* RusConText Benchmark: A Russian Language Evaluation Benchmark for Understanding Context // ACL 2025 Student Research Workshop. 2025. https://aclanthology.org/2025.acl-srw.91/

37. *EleutherAI.* Language Model Evaluation Harness // Zenodo. 2024. v0.4.3. https://zenodo.org/records/10256836

EXPLORING POST-TRAINING QUANTIZATION OF LARGE LANGUAGE MODELS WITH A FOCUS ON RUSSIAN EVALUATION

D. Poimanov[1] [0009-0001-5390-915X], M. Shutov[2] [0009-0009-0530-5034]

[1]*Lomonosov Moscow State University, Moscow, Russia*

[2]*Moscow Institute of Science and Technology, Moscow, Russia*

[1]poimanovdr@my.msu.ru, [2]mihailshutov105@gmail.com

Abstract

The rapid adoption of large language models (LLMs) has made quantization a central technique for enabling efficient deployment under real-world hardware and memory constraints. While English-centric evaluations of low-bit quantization are increasingly available, much less is known about its effects on morphologically rich and resource-diverse languages such as Russian. This gap is particularly important given the recent emergence of high-performing Russian and multilingual LLMs. In this work, we conduct a systematic study of 2-, 3-, and 4-bit post-training quantization (PTQ) for

state-of-the-art Russian LLMs across different model scales (4B and 32B). Our experimental setup covers both standard uniform quantization and specialized low-bit formats, as well as lightweight finetuning for recovery in the most extreme 2-bit setting. Our findings highlight several important trends: (i) the tolerance of Russian LLMs to quantization differs across model families and scales; (ii) 4-bit quantization is generally robust, especially when advanced formats are used; (iii) 3-bit models expose sensitivity to calibration data and scaling strategies; and (iv) 2-bit models, while severely degraded under naive PTQ, can be partially restored through short finetuning. Empirical results show that the model's domain must be considered when using different quantization techniques.

Keywords*: neural networks quantization, compression and optimization of large language models.*

REFERENCES

1. *Shavrina T. et al.* RussianSuperGLUE: A Russian Language Understanding Evaluation Benchmark // Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020. P. 4717–4726. https://doi.org/10.18653/v1/2020.emnlp-main.381

2. *Mendonça J., Lavie A., Trancoso I.* On the Benchmarking of LLMs for Open-Domain Dialogue Evaluation // Proceedings of the 6th Workshop on NLP for Conversational AI (NLP4ConvAI 2024). 2024. P. 1–12. https://doi.org/10.48550/arXiv.2407.03841

3. *Liu J. et al.* Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation //Advances in Neural Information Processing Systems. 2023. Vol. 36. P. 21558–21572. https://doi.org/10.48550/arXiv.2305.01210

4. *Hendrycks D. et al.* Measuring massive multitask language understanding, 2021 // International Conference on Learning Representations. 2021. https://doi.org/10.48550/arXiv.2009.03300

5. *Clark P. et al.* Think you have solved question answering? try arc, the ai2 reasoning challenge // arXiv preprint arXiv:1803.05457. 2018. https://doi.org/10.48550/arXiv.1803.05457

6. *Zellers R. et al.* HellaSwag: Can a Machine Really Finish Your Sentence? // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. P. 4791–4800. https://doi.org/10.48550/arXiv.1905.07830

7. *Dettmers T. et al.* Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale // Advances in neural information processing systems. 2022. Vol. 35, P. 30318–30332. https://doi.org/10.48550/arXiv.2208.07339

8. *Frantar E. et al.* OPTQ: Accurate post-training quantization for generative pre-trained transformers // 11th International Conference on Learning Representations. 2023. https://doi.org/10.48550/arXiv.2210.17323

9. *Lin J. et al.* Awq: Activation-aware weight quantization for on-device llm compression and acceleration // Proceedings of machine learning and systems. 2024. Vol. 6. P. 87–100. https://doi.org/10.1145/3714983.3714987

10. *Xiao G. et al.* Smoothquant: Accurate and efficient post-training quantization for large language models // International conference on machine learning. PMLR, 2023. P. 38087 –38099. https://doi.org/10.48550/arXiv.2211.10438

11. *Tseng A. et al.* Qtip: Quantization with trellises and incoherence processing // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 59597–59620. https://doi.org/10.48550/arXiv.2406.11235

12. *T-Tech.* T-pro-2.0. – Hybrid reasoning model based on Qwen3-32B // HuggingFace.co: The collaboration platform. 2025. URL: https://huggingface.co/t-tech/T-pro-it-2.0

13. *Yandex company.* YandexGPT // HuggingFace.co: The collaboration platform. 2025. URL: https://huggingface.co/yandex/YandexGPT-5-Lite-8B-instruct

14. *Tikhomirov M., Chernyshev D.* Facilitating large language model russian adaptation with learned embedding propagation // Journal of Language and Education. 2024. Vol. 10. No. 4 (40). P. 130–145. https://doi.org/10.48550/arXiv.2412.21140

15. *Team Q. et al.* Qwen2 technical report // arXiv preprint arXiv:2407.10671. 2024. Vol. 2. P. 3. https://doi.org/10.48550/arXiv.2407.10671

16. *Agarwal S. et al.* gpt-oss-120b & gpt-oss-20b Model Card // arXiv e-prints. 2025. P. arXiv: 2508.10925. https://doi.org/10.48550/arXiv.2508.10925

17. *Liu A. et al.* DeepSeek-V3 Technical Report // arXiv e-prints. 2024. P. arXiv: 2412.19437. https://doi.org/10.48550/arXiv.2412.19437

18. *Chee J. et al.* Quip: 2-bit quantization of large language models with guarantees // Advances in Neural Information Processing Systems. 2023. Vol. 36, P. 4396 – 4429. https://doi.org/10.48550/arXiv.2307.13304

19. *Chen M. et al.* Efficientqat: Efficient quantization-aware training for large language models // Annual Meeting of the Association for Computational Linguistics. 2025. Vol. 1. P. 10081–10100. https://doi.org/10.48550/arXiv.2407.11062

20. *Shao W. et al.* OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models // The Twelfth International Conference on Learning Representations. 2024. https://doi.org/10.48550/arXiv.2308.13137

21. *Hu E. J. et al.* Lora: Low-rank adaptation of large language models // International Conference on Machine Learning. 2022. Vol. 1, No. 2. P. 3. https://doi.org/10.48550/arXiv.2106.09685

22. *Han Z. et al.* Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey // arXiv e-prints. 2024. P. arXiv: 2403.14608. https://doi.org/10.48550/arXiv.2403.14608

23. *Egiazarian V. et al.* Extreme compression of large language models via additive quantization // Proceedings of the 41st International Conference on Machine Learning. 2024. P. 12284–12303. https://doi.org/10.48550/arXiv.2401.06118

24. *Tseng A. et al.* QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks // International Conference on Machine Learning. PMLR, 2024. P. 48630–48656. https://doi.org/10.48550/arXiv.2402.04396

25. *Tseng A. et al.* Qtip: Quantization with trellises and incoherence processing // Advances in Neural Information Processing Systems. 2024. Vol. 37. P. 59597–59620. https://doi.org/10.48550/arXiv.2406.11235

26. *Yang A. et al.* Qwen3 technical report // arXiv e-prints. 2025. P. arXiv: 2505.09388. https://doi.org/10.48550/arXiv.2505.09388

27. *Achiam J. et al.* GPT-4 Technical Report // arXiv e-prints. 2023. arXiv: 2303.08774. https://doi.org/10.48550/arXiv.2303.08774

28. *Darvish Rouhani B. et al.* Microscaling data formats for deep learning // arXiv e-prints. 2023. P. arXiv: 2310.10537. https://doi.org/10.48550/arXiv.2310.10537

29. *Weber M. et al.* Redpajama: an open dataset for training large language models // Advances in neural information processing systems. 2024. Vol. 37. P. 116462–116492. https://doi.org/10.52202/079017-3697

30. *Potapov A.* T-Wix – Russian supervised fine-tuning (SFT) dataset // HuggingFace.co: The collaboration platform. 2025. URL: https://huggingface.co/datasets/t-tech/T-Wix

31. *Merity S. et al.* Pointer Sentinel Mixture Models // International Conference on Learning Representations. 2017. https://doi.org/10.48550/arXiv.1609.07843

32. *Korablinov V., Braslavski P.* RuBQ: A Russian dataset for question answering over Wikidata // International Semantic Web Conference. Cham: Springer International Publishing. 2020. P. 97–110. https://doi.org/10.1007/978-3-030-62466-8_7

33. *Li H. et al.* CMMLU: Measuring massive multitask language understanding in Chinese // Findings of the Association for Computational Linguistics. 2024. P. 11260–11285. https://doi.org/10.48550/arXiv.2306.09212

34. *Bisk Y. et al.* Piqa: Reasoning about physical commonsense in natural language // Proceedings of the AAAI conference on artificial intelligence. 2020. Vol. 34. №. 05. P. 7432–7439. https://doi.org/10.1609/aaai.v34i05.6239

35. *Fenogenova A. et al.* MERA: A Comprehensive LLM Evaluation in Russian //Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024. P. 9920–9948. https://doi.org/10.18653/v1/2024.acl-long.534

36. *Chirkin A. et al.* RusConText Benchmark: A Russian Language Evaluation Benchmark for Understanding Context // ACL 2025 Student Research Workshop. 2025. https://aclanthology.org/2025.acl-srw.91/

37. *EleutherAI.* Language Model Evaluation Harness // Zenodo. 2024. v0.4.3. https://zenodo.org/records/10256836

СВЕДЕНИЯ ОБ АВТОРАХ

ПОЙМАНОВ Дмитрий Романович. Окончил магистратуру кафедры математических методов прогнозирования (ММП) факультета вычислительной математики и кибернетики (ВМК) Московского государственного университета имени М.В. Ломоносова в 2024 году. В настоящее время – аспирант ММП ВМК МГУ.

Dmitrii Romanovich POIMANOV. Graduated from the Master's program of the Department of Mathematical Methods of Forecasting (MMP) at the Faculty of Computational Mathematics and Cybernetics (CMC) of Lomonosov Moscow State University in 2024. He is currently a PhD student at MSU.

email: poimanovdr@my.msu.ru

ORCID: 0009-0001-5390-915X

ШУТОВ Михаил Сергеевич. Окончил магистратуру ФПМИ МФТИ в 2024 году. В настоящее время – аспирант ФПМИ МФТИ.

Mikhail Sergeevich SHUTOV. Graduated from the Master's program at the Moscow Institute of Physics and Technology (MIPT), Faculty of Applied Mathematics and Computer Science in 2024. He is currently a Ph.D. student at the same faculty.

email: mihailshutov105@gmail.com

ORCID: 0009-0009-0530-5034

Материал поступил в редакцию 11 октября 2025 года