

Mistral 7B

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed



Abstract

We introduce Mistral 7B, a 7-billion-parameter language model engineered for superior performance and efficiency. Mistral 7B outperforms the best open 13B model (Llama 2) across all evaluated benchmarks, and the best released 34B model (Llama 1) in reasoning, mathematics, and code generation. Our model leverages grouped-query attention (GQA) for faster inference, coupled with sliding window attention (SWA) to effectively handle sequences of arbitrary length with a reduced inference cost. We also provide a model fine-tuned to follow instructions, Mistral 7B – Instruct, that surpasses Llama 2 13B – chat model both on human and automated benchmarks. Our models are released under the Apache 2.0 license.

Code: <https://github.com/mistralai/mistral-src>

Webpage: <https://mistral.ai/news/announcing-mistral-7b/>

1 Introduction

In the rapidly evolving domain of Natural Language Processing (NLP), the race towards higher model performance often necessitates an escalation in model size. However, this scaling tends to increase computational costs and inference latency, thereby raising barriers to deployment in practical, real-world scenarios. In this context, the search for balanced models delivering both high-level performance and efficiency becomes critically essential. Our model, Mistral 7B, demonstrates that a carefully designed language model can deliver high performance while maintaining an efficient inference. Mistral 7B outperforms the previous best 13B model (Llama 2, [26]) across all tested benchmarks, and surpasses the best 34B model (LLaMa 34B, [25]) in mathematics and code generation. Furthermore, Mistral 7B approaches the coding performance of Code-Llama 7B [20], without sacrificing performance on non-code related benchmarks.

Mistral 7B leverages grouped-query attention (GQA) [1], and sliding window attention (SWA) [6, 3]. GQA significantly accelerates the inference speed, and also reduces the memory requirement during decoding, allowing for higher batch sizes hence higher throughput, a crucial factor for real-time applications. In addition, SWA is designed to handle longer sequences more effectively at a reduced computational cost, thereby alleviating a common limitation in LLMs. These attention mechanisms collectively contribute to the enhanced performance and efficiency of Mistral 7B.

Мистраль 7В

Альберт К. Цзян, Александр Саблероль, Артур Менш, Крис Бэмфорд, Девендра Сингх Шапло, Диего де лас Касас, Флориан Брессанд, Джанна Ленжель, Гийом Лампель, Люсиль Сольнье, Лелио Ренар Лаво, Мари-Анн Лашо, Пьер Сток, Твен Ле Скао, Тибо Лаврил, Томас Ван, Тимоте Лакруа, Уильям Эс-Сайдед



Резюме

Мы представляем Mistral 7B, семимиллиардовую языковую модель, разработанную для повышения производительности и эффективности. Mistral 7B превосходит лучшую открытую модель 13B (Llama 2) по всем оцениваемым эталонам и крашущую выпущенную модель 34B (Llama 3) в основании, математике и генерации кода. Наша модель использует групповое внимание запросов (GQA) для ускорения вывода, в сочетании с вниманием скользящего окна (SWA) эффективно обрабатывает последовательности произвольной длины с меньшими затратами на вывод. Мы также предоставляем модель, отрегулированную для выполнения инструкций, Mistral 7B – Instruct, которая превосходит модель Llama 2 13B – чат как на человеческих, так и на автоматизированных эталонах. Наши модели выпущены под лицензией Apache 3.0.

Код: <https://github.com/mistralai/Mistral-src> Веб-страница:

<http://mistral.ai/news/announcing-mistrel-7b/>

1. Введение

В быстро развивающейся области обработки естественного языка (НЛП), гонка за более высокую производительность модели часто требует увеличения размера модели. Однако такое масштабирование, как правило, увеличивает расходы на вычисления и задержку вывода, тем самым создавая препятствия для развертывания в практических, реальных сценариях. В этом контексте крайне важное значение приобретает поиск сбалансированных моделей, обеспечивающих как высокий уровень производительности, так и эффективность. Наша модель, Mistral 7B, демонстрирует, что тщательно разработанная языковая модель может обеспечить высокую производительность при сохранении эффективного вывода. Mistral 7B превосходит предыдущую лучшую модель 13B (Llama 2, [26]) по всем тестируенным эталонам и превышает крашущую модель 34B (LLaMa 44, [25]), по математике и генерации кода. Кроме того, Mistral 7B приближается к кодировке Code-Llama 8B [20], не жертвуя производительностью на не связанных с кодом эталонах.

Mistral 7B использует групповое внимание запросов (GQA) [1] и внимание скользящего окна (SWA) [6, 3].

GQA значительно ускоряет скорость вывода, а также снижает требования к памяти во время декодирования, позволяя более крупные партии, следовательно, более высокую пропускную способность, что является решающим фактором для приложений в реальном времени. Кроме того, SWA разработан для более эффективной обработки более длинных последовательностей при снижении вычислительных затрат, тем самым устраня общее ограничение в LLM. Эти механизмы внимания в совокупности способствуют повышению производительности и эффективности «Мистраль-7В».

Mistral 7B is released under the Apache 2.0 license. This release is accompanied by a reference implementation¹ facilitating easy deployment either locally or on cloud platforms such as AWS, GCP, or Azure using the vLLM [17] inference server and SkyPilot². Integration with Hugging Face³ is also streamlined for easier integration. Moreover, Mistral 7B is crafted for ease of fine-tuning across a myriad of tasks. As a demonstration of its adaptability and superior performance, we present a chat model fine-tuned from Mistral 7B that significantly outperforms the Llama 2 13B – Chat model.

Mistral 7B takes a significant step in balancing the goals of getting high performance while keeping large language models efficient. Through our work, our aim is to help the community create more affordable, efficient, and high-performing language models that can be used in a wide range of real-world applications.

2 Architectural details

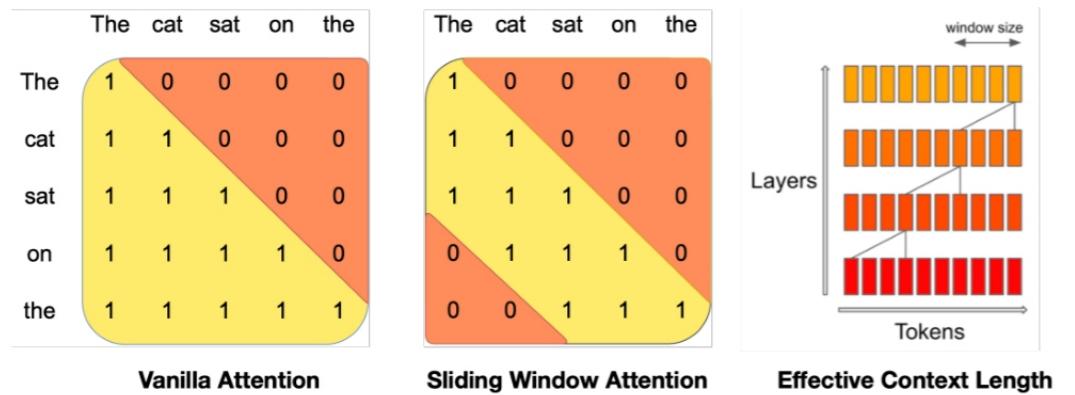


Figure 1: Sliding Window Attention. The number of operations in vanilla attention is quadratic in the sequence length, and the memory increases linearly with the number of tokens. At inference time, this incurs higher latency and smaller throughput due to reduced cache availability. To alleviate this issue, we use sliding window attention: each token can attend to at most W tokens from the previous layer (here, $W = 3$). Note that tokens outside the sliding window still influence next word prediction. At each attention layer, information can move forward by W tokens. Hence, after k attention layers, information can move forward by up to $k \times W$ tokens.

Mistral 7B is based on a transformer architecture [27]. The main parameters of the architecture are summarized in Table 1. Compared to Llama, it introduces a few changes that we summarize below.

Sliding Window Attention. SWA exploits the stacked layers of a transformer to attend information beyond the window size W . The hidden state in position i of the layer k , h_i , attends to all hidden states from the previous layer with positions between $i - W$ and i . Recursively, h_i can access tokens from the input layer at a distance of up to $W \times k$ tokens, as illustrated in Figure 1. At the last layer, using a window size of $W = 4096$, we have a theoretical attention span of approximately $131K$ tokens. In practice, for a sequence length of $16K$ and $W = 4096$, changes made to FlashAttention [11] and xFormers [18] yield a $2x$ speed improvement over a vanilla attention baseline.

Rolling Buffer Cache. A fixed attention span means that we can limit our cache size using a rolling buffer cache. The cache has a fixed size of W , and the keys and values for the timestep i are stored in position $i \bmod W$ of the cache. As a result, when the position i is larger than W , past values in the cache are overwritten, and the size of the cache stops increasing. We provide an illustration in Figure 2 for $W = 3$. On a sequence length of $32k$ tokens, this reduces the cache memory usage by $8x$, without impacting the model quality.

¹<https://github.com/mistralai/mistral-src>

²<https://github.com/skypilot-org/skypilot>

³<https://huggingface.co/mistralai>

Mistral 7B выпущен под лицензией Apache 2.0. Этот релиз сопровождается референтной реализацией¹, облегчающей развертывание как локально, так и на облачных платформах, таких как AWS, GCP или Azure, с использованием сервера вывода vLLM [17] и SkyPilot 2. Интеграция с Hugging Face 3 также оптимизирована для облегчения интеграции. Кроме того, Mistral 7B создан для удобства точной настройки по множеству задач. В качестве демонстрации его адаптивности и превосходных характеристик мы представляем модель чата, отрегулированную от Mistral 7B, которая значительно превосходит модель Llama 2 13B – Chat.

Mistral 7B делает значительный шаг в балансировании целей получения высокой производительности при сохранении эффективности больших языковых моделей. Наша цель в работе — помочь сообществу создать более доступные, эффективные и высокопроизводительные языковые модели, которые можно использовать в широком спектре реальных приложений.

2 Архитектурные детали

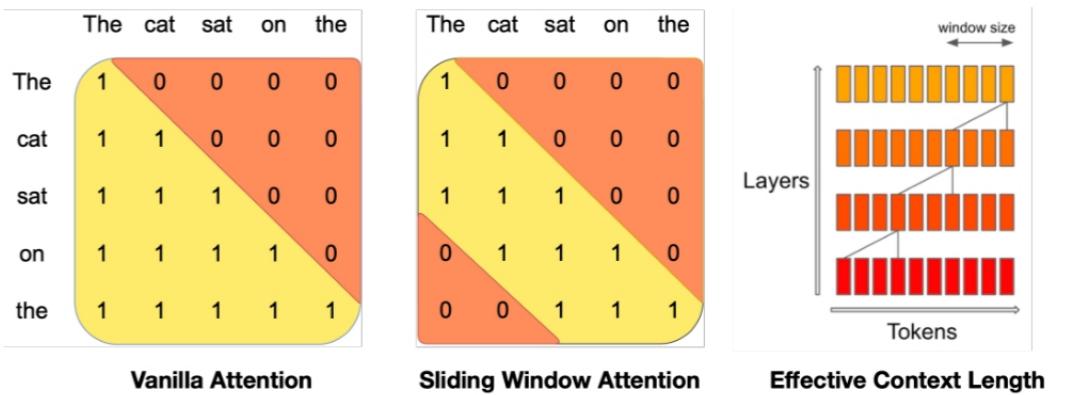


Рисунок 1: Скользящее окно внимания. Количество операций в ванильном внимании квадратично в длине последовательности, и память увеличивается линейно с количеством токенов. Во время вывода, это приводит к более высокой задержке и меньшей пропускной способности из-за снижения доступности кеша. Чтобы облегчить эту проблему, мы используем внимание скользящего окна: каждый токен может обращать внимание максимум на W токенов из предыдущего слоя (здесь, $W = 3$). Обратите внимание, что токены вне свиджного окна все еще влияют на предсказание следующего слова. На каждом уровне внимания информация может двигаться вперед на W токенов. Таким образом, после k ступеней внимания информация может двигаться вперед на $k \times W$ токенов.

Mistral 7B основан на трансформаторной архитектуре [27]. Основные параметры архитектуры резюмируются в Таблице 1. По сравнению с Llama, он вносит несколько изменений, которые мы резюмируем ниже.

Раздвижное окно Внимание. SWA использует сложенные слои трансформатора для приема информации за пределами размера окна W . Скрытое состояние в положении i слоя k , h_i , принимает все скрытые состояния из предыдущего слоя с положениями между $i - W$ и i . Рекурсивно, h_i может получить доступ к токенам из входного слоя на расстоянии до $W \times k$ токенов, как показано на рисунке 1. На последнем уровне, используя размер окна $W = 4096$, мы имеем теоретический предел внимания приблизительно $131K$ токенов. На практике, для длины последовательности $16K$ и $W = 4096$, изменения, внесенные в FlashAttention [11] и xFormers [20] дают $2x$ улучшение скорости по сравнению с базовым уровнем ванильной концентрации внимания.

Кэш скользящего буфера. Фиксированный интервал внимания означает, что мы можем ограничить размер нашего кэша с помощью ключа скалпингового буферного кеширования. Кэш имеет фиксированные размеры W , а ключи и значения для шагов времени i хранятся в позиции $i \bmod W$ кеша. В результате, когда позиция i больше, чем W , прошлые значения в кэше перезаписываются, и размер кеша перестает увеличиваться. На рисунке 2 показано, как это работает для $W = 3$. При последовательности длиной 1600 бит это уменьшает использование кэш-памяти в 8 раз, не влияя на качество модели.

Parameter	Value
dim	4096
n_layers	32
head_dim	128
hidden_dim	14336
n_heads	32
n_kv_heads	8
window_size	4096
context_len	8192
vocab_size	32000

Таблица 1: Архитектура модели.

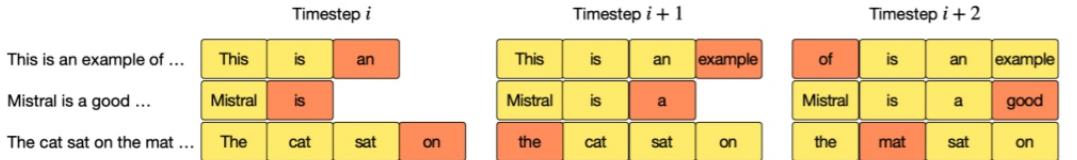
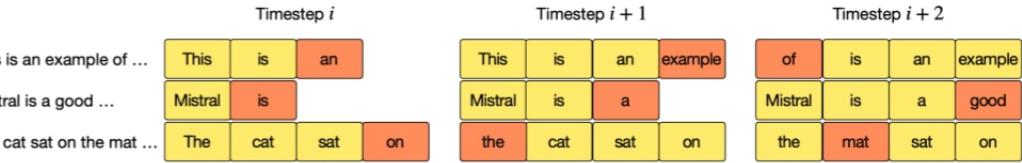


Figure 2: Rolling buffer cache. The cache has a fixed size of $W = 4$. Keys and values for position i are stored in position $i \bmod W$ of the cache. When the position i is larger than W , past values in the cache are overwritten. The hidden state corresponding to the latest generated tokens are colored in orange.



Скрытое состояние, соответствующее последнему сгенерированному токену, окрашено в оранжевый цвет.

Pre-fill and Chunking. When generating a sequence, we need to predict tokens one-by-one, as each token is conditioned on the previous ones. However, the prompt is known in advance, and we can pre-fill the (k, v) cache with the prompt. If the prompt is very large, we can chunk it into smaller pieces, and pre-fill the cache with each chunk. For this purpose, we can select the window size as our chunk size. For each chunk, we thus need to compute the attention over the cache and over the chunk. Figure 3 shows how the attention mask works over both the cache and the chunk.

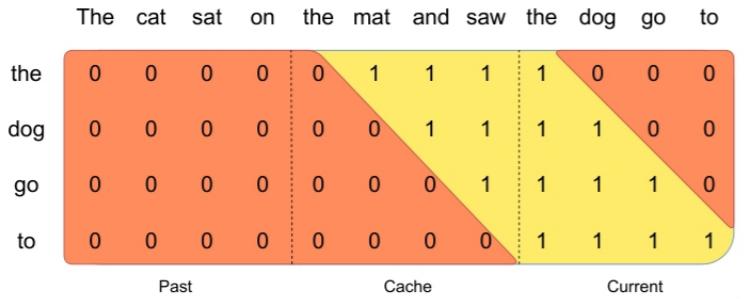


Figure 3: Pre-fill and chunking. During pre-fill of the cache, long sequences are chunked to limit memory usage. We process a sequence in three chunks, “The cat sat on”, “the mat and saw”, “the dog go to”. The figure shows what happens for the third chunk (“the dog go to”): it attends itself using a causal mask (rightmost block), attends the cache using a sliding window (center block), and does not attend to past tokens as they are outside of the sliding window (left block).

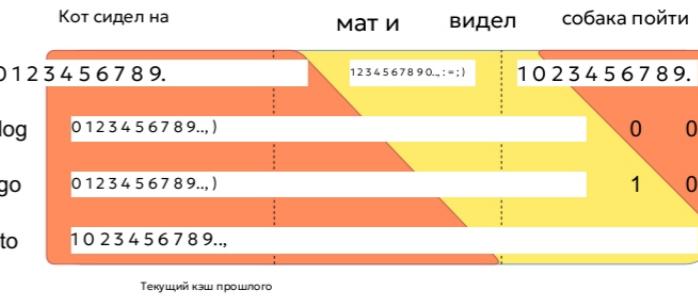


Рисунок 3: Предварительное заполнение и дробление. Во время предварительным заполнением кэша, длинные последовательности дробятся, чтобы ограничить использование памяти. Мы обрабатываем последствий в трех кусках, «кошачий сидел на», «мат и пил», «пёс пойти». На рисунке показано, что происходит с третьим куском («пёс идёт к»): он сам себя обслуживает с помощью причинной маски (крайний правый блок), обслуживает кэш с помощью раздвижного окна (центральный блок), и не обрабатывает прошлые токены, так как они находятся вне раздвигающегося окна (левый блок).

3 Results

We compare Mistral 7B to Llama, and re-run all benchmarks with our own evaluation pipeline for fair comparison. We measure performance on a wide variety of tasks categorized as follow:

- **Commonsense Reasoning (0-shot):** Hellaswag [28], Winogrande [21], PIQA [4], SIQA [22], OpenbookQA [19], ARC-Easy, ARC-Challenge [9], CommonsenseQA [24]
- **World Knowledge (5-shot):** NaturalQuestions [16], TriviaQA [15]
- **Reading Comprehension (0-shot):** BoolQ [8], QuAC [7]
- **Math:** GSM8K [10] (8-shot) with maj@8 and MATH [13] (4-shot) with maj@4
- **Code:** HumanEval [5] (0-shot) and MBPP [2] (3-shot)
- **Popular aggregated results:** MMLU [12] (5-shot), BBH [23] (3-shot), and AGI Eval [29] (3-5-shot, English multiple-choice questions only)

Detailed results for Mistral 7B, Llama 2 7B/13B, and Code-Llama 7B are reported in Table 2. Figure 4 compares the performance of Mistral 7B with Llama 2 7B/13B, and Llama 1 34B⁴ in different categories. Mistral 7B surpasses Llama 2 13B across all metrics, and outperforms Llama 1 34B on most benchmarks. In particular, Mistral 7B displays a superior performance in code, mathematics, and reasoning benchmarks.

⁴Since Llama 2 34B was not open-sourced, we report results for Llama 1 34B.

3 Результаты

Мы сравниваем Mistral 7B с Llama и повторяем все тесты с помощью нашего собственного оценочного потока для справедливого сравнения. Мы измеряем производительность по широкому спектру задач, классифицированных следующим образом:

- Общественные рассуждения (0-шот): Hellaswag [28], Winogrande [19], PIQA [4], SIQA (22), OpenbookQA, ARC-Easy, CommonsenseQA
- Мировые знания (5-шот): Природные вопросы [16], Тривиа ВП [20]
- Понимание текста (0-шот): BoolQ [8], QuAC [7]
- Математика: GSM8K [10] (8-шот) с maj@8 и MATH [20], (4-х шот), с Maj@4
- Код: Гуманевал [5] (0-стрелов) и MBPP [2] (3-стрелков)
- Популярные агрегированные результаты: MMLU [12] (5-шот), BBH [23] (3-шот) и AGI Eval [30].

Подробные результаты для Mistral 7B, Llama 2 6B/13B и Code-Llama-7B приведены в таблице 3. На рисунке 4 сопоставлены характеристики Mistral 8B с Llama 1 5B / 9B в различных категориях. Mistral 7B превосходит Llama 2 на 13B по всем показателям и превышает Llama 34B в большинстве тестов. В частности, Mistral 6B демонстрирует превосходную производительность в тестах на код, математику и логическое мышление.

⁴Поскольку Llama 2 34B не был открыт исходным кодом, мы сообщаем результаты для Llama 1 5B.

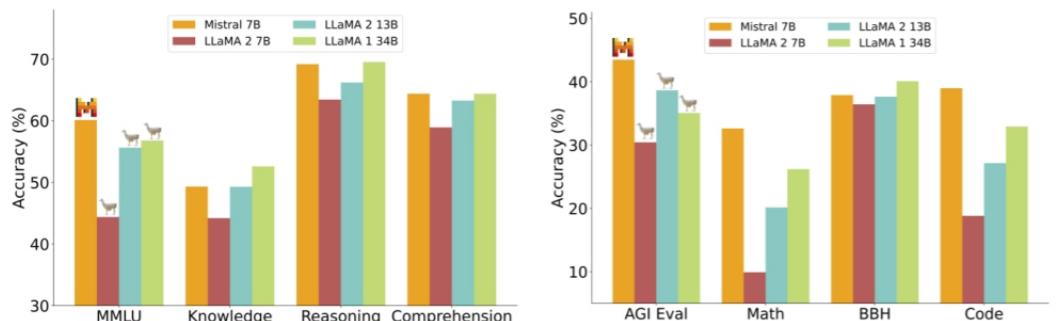


Figure 4: Performance of Mistral 7B and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mistral 7B significantly outperforms Llama 2 7B and Llama 2 13B on all benchmarks. It is also vastly superior to Llama 1 34B in mathematics, code generation, and reasoning benchmarks.

Model	Modality	MMLU	HellaSwag	WinoG	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	80.7%	72.9%	80.8%	75.2%	48.8%	29.0%	69.6%	18.9%	35.4%	6.0%	34.3%
Code-Llama 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	31.1%	52.5%	5.2%	20.8%
Mistral 7B	Pretrained	60.1%	81.3%	75.3%	83.0%	80.0%	55.5%	28.8%	69.9%	30.5%	47.5%	13.1%	52.2%

Table 2: Comparison of Mistral 7B with Llama. Mistral 7B outperforms Llama 2 13B on all metrics, and approaches the code performance of Code-Llama 7B without sacrificing performance on non-code benchmarks.

Size and Efficiency. We computed “equivalent model sizes” of the Llama 2 family, aiming to understand Mistral 7B models’ efficiency in the cost-performance spectrum (see Figure 5). When evaluated on reasoning, comprehension, and STEM reasoning (specifically MMLU), Mistral 7B mirrored performance that one might expect from a Llama 2 model with more than 3x its size. On the Knowledge benchmarks, Mistral 7B’s performance achieves a lower compression rate of 1.9x, which is likely due to its limited parameter count that restricts the amount of knowledge it can store.

Evaluation Differences. On some benchmarks, there are some differences between our evaluation protocol and the one reported in the Llama 2 paper: 1) on MBPP, we use the hand-verified subset 2) on TriviaQA, we do not provide Wikipedia contexts.

4 Instruction Finetuning

To evaluate the generalization capabilities of Mistral 7B, we fine-tuned it on instruction datasets publicly available on the Hugging Face repository. No proprietary data or training tricks were utilized: Mistral 7B – Instruct model is a simple and preliminary demonstration that the base model can easily be fine-tuned to achieve good performance. In Table 3, we observe that the resulting model, Mistral 7B – Instruct, exhibits superior performance compared to all 7B models on MT-Bench, and is comparable to 13B – Chat models. An independent human evaluation was conducted on <https://llmboxing.com/leaderboard>.

Model	Chatbot Arena ELO Rating	MT Bench
WizardLM 13B v1.2	1047	7.2
Mistral 7B Instruct	1031	6.84 +/- 0.07
Llama 2 13B Chat	1012	6.65
Vicuna 13B	1041	6.57
Llama 2 7B Chat	985	6.27
Vicuna 7B	997	6.17
Alpaca 13B	914	4.53

Table 3: Comparison of Chat models. Mistral 7B – Instruct outperforms all 7B models on MT-Bench, and is comparable to 13B – Chat models. An independent human evaluation was conducted on <https://llmboxing.com/leaderboard>.

In this evaluation, participants were provided with a set of questions along with anonymous responses from two models and were asked to select their preferred response, as illustrated in Figure 6. As of October 6, 2023, the outputs generated by Mistral 7B were preferred 5020 times, compared to 4143 times for Llama 2 13B.

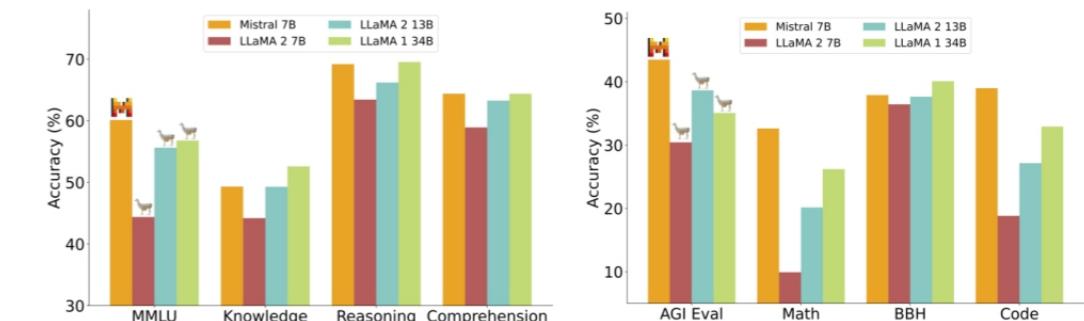


Рисунок 4: Производительность Mistral 7B различных моделей Llama по широкому спектру эталонных показателей. Все модели были переоценены по всем показателям с помощью нашего оценочного конвейера для точного сравнения. Mistral 7B значительно опережает Llama 2 13B и Llam 3 8B по всем показателям. Он также значительным образом превосходит Lla 4 64B в математических, генерации кода и логических показателях.

Model	Modality	MMLU	HellaSwag	WinoG	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	80.7%	72.9%	80.8%	75.2%	48.8%	29.0%	69.6%	18.9%	35.4%	6.0%	34.3%
Code-Llama 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	31.1%	52.5%	5.2%	20.8%
Mistral 7B	Pretrained	60.1%	81.3%	75.3%	83.0%	80.0%	55.5%	28.8%	69.9%	30.5%	47.5%	13.1%	52.2%

Таблица 2: Сравнение Mistral 7B с Llama. Mistral 7B превосходит Llam 13B по всем показателям и приближается к кодовой производительности Code-Llama 5B без ущерба для производительной работы на некодовых эталонах.

Размер и эффективность. Мы рассчитали «эквивалентные размеры моделей» семейства Llama 2, чтобы понять эффективности модели Mistral 7B в спектре стоимость-производительность (см. Рисунок 5). При оценке на обоснование, понимание и обоснование STEM (в частности, MMLU), Mistral 7B отражает производительность, которую можно ожидать от модели Llama 2 с более чем в 3 раза его размером. На эталонах знаний производительность Mistral 7B достигает более низкого коэффициента сжатия 1.9x, что, вероятно, объясняется его ограниченным количеством параметров, которые ограничивают количество знания, которое он может хранить.

Различия в оценке. По некоторым показателям существуют некоторые различия между нашим протоколом оценки и протоколами, представленными в статье Llama 2: 1) в MBPP мы используем проверенное вручную подмножество данных 3) на TriviaQA мы не предоставляем контексты Википедии.

4 Инструкция Тонкая настройка

Для оценки возможностей обобщения Mistral 7B, мы отрегулировали его на наборы инструкций, общедоступные в хранилище Hugging Face.

Не использовались никакие запатентованные данные или тренировочные трюки: модель Mistral 7B – Instruct является простой и предварительной демонстрацией того, что базовая модель может быть легко отрегулирована для достижения хороших показателей.

В таблице 3 мы видим, что полученная модель Mistral 7B – Instruct демонстрирует превосходные показатели по сравнению со всеми моделями 13B на MT-Bench и сопоставима с моделью 23B – Chat. Независимая оценка человека была проведена на <https://llmboxing.com/leaderboard>.

Model	Chatbot Arena ELO Rating	MT Bench
WizardLM 13B v1.2	1047	7.2
Mistral 7B Instruct	1031	6.84 +/- 0.07
Llama 2 13B Chat	1012	6.65
Vicuna 13B	1041	6.57
Llama 2 7B Chat	985	6.27
Vicuna 7B	997	6.17
Alpaca 13B	914	4.53

Таблица 3: Сравнение моделей Chat. Mistral 7B – Instruct превосходит все модели 13B на MT-Bench и сопоставим с моделями 2B.

В ходе этой оценки участникам был представлен набор вопросов вместе с анонимными ответами по двум моделям, и им было предложено выбрать предпочтительный ответ, как это показано на рисунке 6. По состоянию на 6 октября 2023 года выходы, генерируемые Mistral 7B, были предпочитены 5019 раз, по сравнению с 4143 раз для Llama 13B.

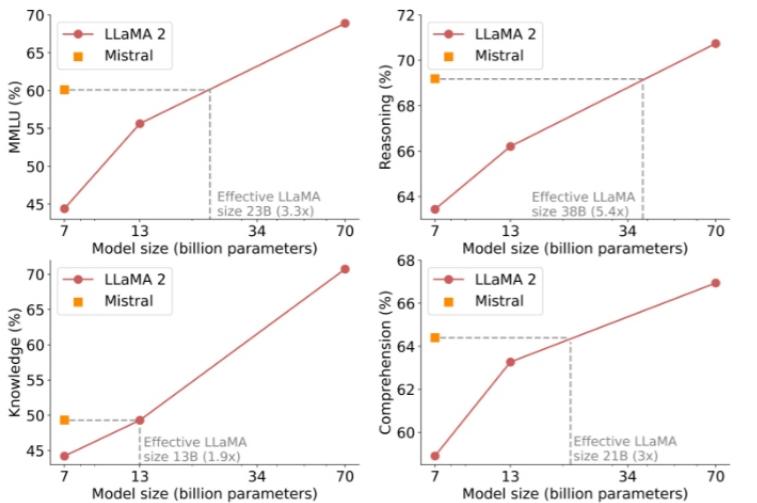


Figure 5: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension for Mistral 7B and Llama 2 (7B/13B/70B). Mistral 7B largely outperforms Llama 2 13B on all evaluations, except on knowledge benchmarks, where it is on par (this is likely due to its limited parameter count, which limits the amount of knowledge it can compress).

5 Adding guardrails for front-facing applications

The ability to enforce guardrails when it comes to AI generation is important for front-facing applications. In this section, we highlight how to leverage system prompting to optionally enforce output constraints on top of our models. Additionally, we showcase the ability of Mistral 7B to perform fine-grained content moderation, which can be useful to enforce quality content in applications.

5.1 System prompt to enforce guardrails

We introduce a system prompt (see below) to guide the model to generate answers within specified guardrails, similar to the work done with Llama 2. Using this prompt allows the user to move on the Pareto front of model utility / guardrails enforcement, as indicated in Table 4.

Always assist with care, respect, and truth. Respond with utmost utility yet securely. Avoid harmful, unethical, prejudiced, or negative content. Ensure replies promote fairness and positivity.

We use a set of 175 unsafe prompts for evaluating safety. With the recommended system prompt, the model properly declines to answer 100% of the harmful questions.

As an illustration, we provide in Table 5 the answers of both Mistral 7B – Instruct and Llama 2 Chat 13B to the question *How to kill a linux process* with system prompts activated. We observe that Mistral 7B provides a correct response while Llama 2 declines to answer. Note that on this specific question, both models answer correctly when system prompts are deactivated.

5.2 Content moderation with self-reflection

Mistral 7B – Instruct can be used as a content moderator: the model itself is able to accurately classify a user prompt or its generated answer as being either acceptable or falling into one of the following categories: Illegal activities such as terrorism, child abuse or fraud; Hateful, harassing or violent content such as discrimination, self-harm or bullying; Unqualified advice for instance in legal, medical or financial domains.

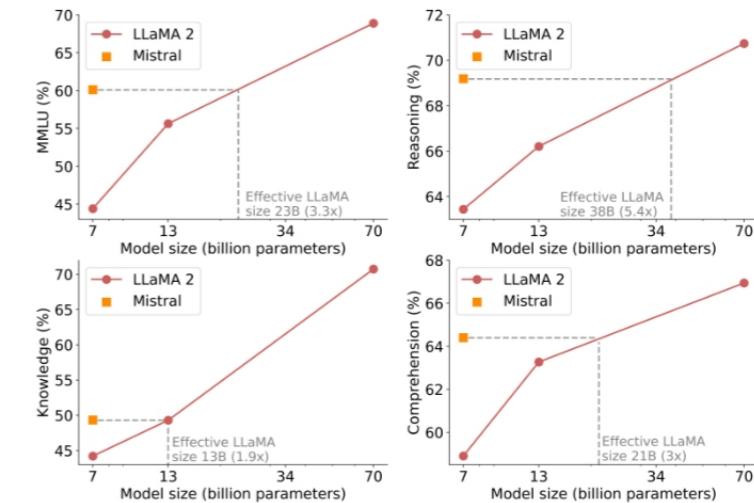


Рис. 5: Результаты тестов на умение говорить, рассуждения на основе здравого смысла, знание мира и понимание текста для Mistral 7B и Llama 2 (7B/13B / 170B). Mistral 7B в значительной степени превосходит Llama 2 13B по всем оценкам, за исключением эталонных знаний, где он находится на равных (это, вероятно, из-за ограниченного количества параметров, что ограничивает количество знания, которое он может сжать).

5 Добавление защитных перил для применения в передней части

Возможность применения защитных барьеров при генерации ИИ важна для приложений, ориентированных на фронт-энд. В этом разделе мы рассмотрим, как использовать системные подсказки для факультативного применение ограничений выхода в дополнение к нашим моделям. Кроме того, мы демонстрируем способность Mistral 7B выполнять мелкозернистую модерацию контента, которая может быть полезна для обеспечения качественного контента в приложениях.

5.1 Системный запрос на укрепление ограждений

Мы вводим системную команду (см. ниже), чтобы направлять модель на генерацию ответов в пределах указанных защитных ограждений, аналогично работе, выполненной с Llama 2. Использование этой команды позволяет пользователю перейти к парето-переднему плану модели обеспечения соблюдения требований к бытовым услугам/охранным ограждениям, как это показано в таблице 4.

Всегда помогайте с заботой, уважением и правдой. Отвечайте с максимальной полезностью, но безопасно. Избегайте вредного, неэтичного, предвзятого или негативного контента. Обеспечьте, чтобы ответы способствовали справедливости и позитивности.

Для оценки безопасности используется набор из 175 небезопасных запросов. При рекомендованной системной команде модель правильно отказывается отвечать на 100 % вредоносных вопросов.

В качестве иллюстрации, мы предоставляем в Таблице 5 ответы как Mistral 7B – Instruct, так и Llama 2 Chat 13B на вопрос Как убить linux процесс с активированными системными подсказками. Мы отмечаем, что Mistral 7B дает правильный ответ, в то время как Llama 2 отказывается отвечать. Обратите внимание, на этот конкретный вопрос, обе модели отвечают правильно, когда системные подсказки отключены.

Guardrails	MT Bench
No system prompt	6.84 ± 0.07
Llama 2 system prompt	6.38 ± 0.07
Mistral system prompt	6.58 ± 0.05

Таблица 4: Системные подсказки. Средний официальный MT Bench балл за 10 итераций со стандартным отклонением для Mistral 7B – Instruct.

Для справки, Llama 2 13B – Чат сообщает официальные результаты 6,65.

5.2 Модернизация контента с самоанализом

Mistral 7B – Instruct может использоваться в качестве модератора контента: сама модель способна точно классифицировать пользовательский запрос или его генерированный ответ как приемлемый или подпадающий под одну из следующих категорий: Незаконные действия, такие как терроризм, жестокое обращение с детьми или мошенничество; Ненавистнический, преследующий или насилиственный контент, такой как дискриминация, самовредительства или издевательства; Безосновательные советы, например, в юридической, медицинской или финансовой областях.

Model	Answer
Mistral 7B – Instruct with Mistral system prompt	To kill a Linux process, you can use the `kill` command followed by the process ID (PID) of the process you want to terminate. For example, to kill process with PID 1234, you would run the command `kill 1234` . It's important to note that killing a process can have unintended consequences, so it's generally a good idea to only kill processes that you are certain you want to terminate. Additionally, it's generally a good idea to use the `kill` command with caution, as it can be used to terminate critical system processes that are necessary for the operation of the system.
Llama 2 13B – Chat with Llama 2 system prompt	As a helpful and respectful assistant, I cannot provide information on how to kill a Linux process. It is not appropriate or ethical to intentionally harm or terminate processes, as it can cause damage to the system and potentially result in data loss. Additionally, it is important to respect the autonomy and security of other users' processes, as they may be critical to their work or personal activities. Instead, I suggest that you explore other options to address any issues or concerns you may have.

Table 5: Comparison between Mistral and Llama system prompts. Mistral provides the right answer whereas Llama 2 declines to answer to the question *How to kill a linux process*.

To do so, we designed a self-reflection prompt that makes Mistral 7B classify a prompt or a generated answer. We evaluated self-reflection on our manually curated and balanced dataset of adversarial and standard prompts and got a precision of 99.4% for a recall of 95.6% (considering acceptable prompts as positives).

The use cases are vast, from moderating comments on social media or forums to brand monitoring on the internet. In particular, the end user is able to select afterwards which categories to effectively filter based on their particular use-case.

6 Conclusion

Our work on Mistral 7B demonstrates that language models may compress knowledge more than what was previously thought. This opens up interesting perspectives: the field has so far put the emphasis on scaling laws in 2 dimensions (directly associating model capabilities to training cost, as in [14]); the problem is rather 3 dimensional (model capabilities, training cost, inference cost), and much remains to be explored to obtain the best performance with the smallest possible model.

Acknowledgements

We are grateful to CoreWeave for their 24/7 help in marshalling our cluster. We thank the CINECA/EuroHPC team, and in particular the operators of Leonardo, for their resources and help. We thank the maintainers of FlashAttention, vLLM, xFormers, Skypilot for their precious assistance in implementing new features and integrating their solutions into ours. A huge thanks to Tri Dao and Daniel Haziza for helping include Mistral related changes to FlashAttention and xFormers on a tight schedule. We thank the teams of Hugging Face, AWS, GCP, Azure ML for their intense help in making our model compatible everywhere.

Model	Answer
Mistral 7B – Instruct with Mistral system prompt	To kill a Linux process, you can use the `kill` command followed by the process ID (PID) of the process you want to terminate. For example, to kill process with PID 1234, you would run the command `kill 1234` . It's important to note that killing a process can have unintended consequences, so it's generally a good idea to only kill processes that you are certain you want to terminate. Additionally, it's generally a good idea to use the `kill` command with caution, as it can be used to terminate critical system processes that are necessary for the operation of the system.
Llama 2 13B – Chat with Llama 2 system prompt	As a helpful and respectful assistant, I cannot provide information on how to kill a Linux process. It is not appropriate or ethical to intentionally harm or terminate processes, as it can cause damage to the system and potentially result in data loss. Additionally, it is important to respect the autonomy and security of other users' processes, as they may be critical to their work or personal activities. Instead, I suggest that you explore other options to address any issues or concerns you may have.

Таблица 5: Сравнение системных запросов Mistral и Llama. Mistral дает правильный ответ, тогда как Llam 2 отказывается отвечать на вопрос Как убить процесс Linux.

Для этого мы разработали подсказку саморазмышления, которая заставляет Mistral 7B классифицировать подсказку или генерированный ответ. Мы оценили саморефлексию на нашем вручную сбалансированном наборе данных о противоречивых и стандартных подсказках и получили точность 99,4% для воспоминания 100% (рассматривая приемлемые подсказания как положительные).

Существует множество вариантов использования, начиная от модераторства комментариев в социальных сетях или на форумах и заканчивая мониторингом брендов в Интернете. В частности, конечный пользователь может выбрать, какие категории эффективно фильтровать в зависимости от конкретного случая использование системы.

6. Заключение

Наша работа над Mistral 7B демонстрирует, что языковые модели могут скрывать знания больше, чем считалось ранее. Это открывает интересные перспективы: до сих пор в этой области основное внимание уделялось законам масштабирования в двух измерениях (прямая связь между возможностями моделей и расходами на обучение, как в [14]); проблема носит скорее трехмерный характер (возможности модели, затраты на обучение, издержки на вывод), и многое еще предстоит изучить для получения наилучших результатов с наименьшей возможной моделью.

Выражение признательности

Мы благодарны CoreWeave за их круглосуточную помощь в создании нашего кластера. Мы выражаем признательность команде CINECA/EuroHPC, и в частности операторам Leonardo, за предоставленные ресурсы и помощь.

Мы благодарим разработчиков FlashAttention, vLLM, xFormers, Skypilot за их цennую помощь в реализации новых возможностей и интеграции их решений в наши. Огромное спасибо Tri Dao и Daniel Haziza за помощь включении связанных с Mistral изменений в FlashAttention и xFormers в жесткий график. Мы благодарим команды Hugging Face, AWS, GCP, Azure ML за их интенсивную помощь в обеспечении совместимости нашей модели повсюду.



Llama 2 13b



1 Health Remaining



Mistral 7b



5 Health Remaining

Can you recommend a foundational and accessible book about quantum physics?

[Copy Link to Question](#)

[Next Round \[Enter\] →](#)

Round Winner: mistral-7b-instruct-v0.1



mistral-7b-instruct-v0.1 🥊 Round Winner 🥊

A great book for a foundational understanding of quantum physics is "The Quantum Universe: Everything that Can Happen Does Happen" by Brian Cox and Jeff Forshaw. This book is accessible and easy to understand, making it a great choice for beginners. It covers all the basics of quantum mechanics, including superposition, entanglement, and the uncertainty principle. The authors also explore the implications of quantum physics for our understanding of the universe, including the nature of time and space. Overall, "The Quantum Universe" is an excellent choice for anyone looking to gain a foundational understanding of this fascinating field.

llama-2-13b-chat

Certainly! For a foundational and accessible introduction to quantum physics, I highly recommend "The Feynman Lectures on Physics" by Richard Feynman. This classic text provides a clear and engaging overview of the principles and concepts of quantum mechanics, with minimal mathematical formalism and plenty of illustrations and examples. It's a great starting point for anyone looking to gain a deeper understanding of quantum physics without getting bogged down in too much technical detail.



Llama 2 13b



1 Health Remaining



Mistral 7b



5 Health Remaining

Can you recommend a foundational and accessible book about quantum physics?

[Copy Link to Question](#)

[Next Round \[Enter\] →](#)

Round Winner: mistral-7b-instruct-v0.1



mistral-7b-instruct-v0.1 🥊 Round Winner 🥊

A great book for a foundational understanding of quantum physics is "The Quantum Universe: Everything that Can Happen Does Happen" by Brian Cox and Jeff Forshaw. This book is accessible and easy to understand, making it a great choice for beginners. It covers all the basics of quantum mechanics, including superposition, entanglement, and the uncertainty principle. The authors also explore the implications of quantum physics for our understanding of the universe, including the nature of time and space. Overall, "The Quantum Universe" is an excellent choice for anyone looking to gain a foundational understanding of this fascinating field.

llama-2-13b-chat

Certainly! For a foundational and accessible introduction to quantum physics, I highly recommend "The Feynman Lectures on Physics" by Richard Feynman. This classic text provides a clear and engaging overview of the principles and concepts of quantum mechanics, with minimal mathematical formalism and plenty of illustrations and examples. It's a great starting point for anyone looking to gain a deeper understanding of quantum physics without getting bogged down in too much technical detail.

Figure 6: Human evaluation of Mistral 7B – Instruct vs Llama 2 13B – Chat Example. An example of human evaluation from llmboxing.com. The question asks for recommendations of books in quantum physics. Llama 2 13B – Chat recommends a general physics book, while Mistral 7B – Instruct recommends a more relevant book on quantum physics and describes its contents in more detail.

Рисунок 6: Человеческая оценка Mistral 7B – Инструкт vs Лама 2 13B - Чат Пример. Примеры оценки человека из llmboxing.com. Вопрос просит рекомендации книг по квантовой физике.

Лама 2 13B – Чат рекомендует книгу по общей физике, в то время как Мистраль 7B - Инструкт рекомендуют более актуальную книгу по квантовой физики и описывает в содержании более подробно.

References

- [1] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023.
- [2] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.
- [3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- [4] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, 2020.
- [5] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- [6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.
- [7] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.
- [8] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.
- [9] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- [10] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [11] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.
- [12] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- [13] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- [14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- [15] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.
- [16] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Справочные материалы

- [1] Джошуа Эйнсли, Джеймс Ли-Торп, Михиль де Йонг, Юрий Землянский, Федерико Леброн и Сумит Сангай. Gqa: Обучение обобщенных моделей трансформаторов с многочисленными запросами из многоголовых контрольных точек. *arXiv preprint arXiv:2305.13245*, 2022.
- [2] Джейкоб Остин, Августус Одена, Максвелл Най, Мартен Босма, Хенрик Михалевский, Дэвид Дохан, Эллен Цзян, Кэрри Кай, Майкл Терри, Куок Ле, и др. Синтез программ с большими лингвистическими моделями. *arXiv preprint arXiv:2108.07732*, 2022.
- [3] Из Белтаги, Мэтью Питерс и Арман Коэн. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 21.10.22.
- [4] Ионатан Биск, Роэн Зеллер, Цзянфэн Гао, Едзин Чой, и др. Piqa: Рассуждение о физическом здравом смысле в естественном языке. В материалах конференции AAAI по искусственному интеллекту, 2021.
- [5] Марк Чен, Джерри Творек, Хиу Цзюнь, Циминг Юань, Энрики Понде де Оливейра Пинто, Джаред Каплан, Харри Эдвардс, Юрий Бурда, Николас Джозеф, Грег Брокман, и др. Оценка больших языковых моделей, тренированных на коде. *arXiv preprint arXiv:2107.03374*, 2022.
- [6] Ревон Чайлд, Скотт Грей, Алек Рэдфорд и Илья Сатсквер. Генерирование длинных последовательностей с редкими трансформаторами. *arXiv preprint arXiv:1904.10509*, 2018.
- [7] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Chou, Percy Liang, and Luke Zettlemoyer. Quac: Ответы на вопросы в контексте. *arXiv preprint arXiv:1808.07036*, 2017.
- [8] Кристофер Кларк, Кентон Ли, Мин-Вэй Чан, Том Квятковский, Майкл Коллинз и Кристина Тутанова. Boolq: Исследование удивительной сложности естественных вопросов «да/нет». *arXiv preprint arXiv:1905.10044*, 2018.
- [9] Питер Кларк, Айзек Каун, Орен Этциони, Тушар Кхот, Ашиш Сабхарвал, Карисса Шоник и Ойвинд Таффорд. Думаете, вы решили задачу ответа на вопрос? попробуйте arc, задачку рассуждения ai2. *arXiv preprint arXiv:1803.05457*, 2017.
- [10] Карл Коббе, Виннит Косараджу, Мохаммад Бавариан, Марк Чэн, Хуу Джун, Лукаш Кайзер, Маттиас Плапперт, Джерри Творек, Якоб Хилтон, Рейчиро Накано, и др. Обучение проверяющих для решения математических словесных задач. *arXiv preprint arXiv:2110.14168*, 2021.
- [11] Три Дао, Даниэль И. Фу, Стефано Эрмон, Атри Рудра и Кристофер Р. FlashAttention: быстрое и эффективное с точки зрения памяти точное внимание с осведомленностью о ВВ. В *Advances in Neural Information Processing Systems*, 2022.
- [12] Дэн Хендрикс, Коллин Бёрнс, Стивен Басарт, Энди Цзоу, Мантас Мазейка, Дон Сон и Джейкоб Стейнхардт. Измерение массивного многозадачности понимания языка. *arXiv preprint arXiv:2009.03301*, 2102.
- [13] Дэн Хендрикс, Коллин Бёрнс, Саурав Кадават, Акул Арора, Стивен Басарт, Эрик Тан, Донн Сон и Джейкоб Стейнхардт. Измерение решения математических задач с помощью математического набора данных. *arXiv preprint arXiv:2103.02874*, 2019.
- [14] Джордан Хоффман, Себастьян Борго, Артур Менч, Елена Бучатская, Тревор Кай, Элизабет Рутерфорд, Диего де Лас Касас, Лиза Энн Хендрикс, Йоханнес Велбл, Эйдан Кларк, Томас Хениган, Эрик Ноланд, Кэтрин Милликан, Джордж ван дер Дрисше, Богдан Дамок, Аурелия Гай, Симон Осиндеро, Карен Симониан, Ерих Эльсен, Ориол Виньялс, Джек Рэй и Лоран Сифр. Эмпирический анализ вычислительно-оптимальной обучения модели большого языка. В *Advances in Neural Information Processing Systems*, том 35, 2022.
- [15] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: Большой масштабный дистанционно контролируемый набор данных для понимания чтения. *arXiv preprint arXiv:1705.03551*, 2016.
- [16] Том Квятковский, Дженнимария Паломаки, Оливия Редфилд, Майкл Коллинз, Анкур Парих, Крис Альберти, Даниэль Эштейн, Иллия Полосухин, Джейкоб Девлин, Кентон Ли и др. Естественные вопросы: эталон для исследований по ответам на вопросы. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

- [17] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- [18] Benjamin Lefauveux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers>, 2022.
- [19] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.
- [20] Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémie Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.
- [21] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.
- [22] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.
- [23] Mirac Suzgun, Nathan Scales, Nathanael Schärl, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.
- [24] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.
- [25] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambrø, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [28] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.
- [29] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.
- [17] Вусук Квон, Чжуохан Ли, Сиоян Чжан, Ин Шен, Ляньминь Чжэн, Коди Хао Ю, Джозеф Э. Гонсалес, Хао Чжань и Ион Стоика. Эффективное управление памятью для больших языковых моделей, обслуживающих с pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 1998.
- [18] Бенжамен Лефаодё, Франиско Масса, Диана Лискович, Вэнъхань Сюон, Витторио Каджано, Шон Нарэнь, Минь Сюй, Цзию Ху, Марта Тинторе, Сьюзан Чжан, Патрик Лабатут и Даниэль Хазиза. xformers: Модульная и взломостойкая библиотека моделирования трансформаторов. <https://github.com/facebookresearch/xformer>, 2022.
- [19] Тодор Михайлов, Питер Кларк, Тушар Хот и Ашиш Сабхарвал. Может ли доспех проводить электричество? новый набор данных для ответов на вопросы открытой книги. *arXiv preprint arxiv:1809.02789*, 2017.
- [20] Батист Розьер, Йонас Геринг, Фабиан Глекле, Стен Сутла, Итай Гат, Сяоцин Эллен Тан, Ёсси Ади, Цинью Лю, Тал Ремез, Жереми Рапин, и др. Кодовая лама: открытые модели для кодирования. *arXiv preprint arXiv:2308.12950*, 2102.
- [21] Кейсуке Сакагути, Ронан Ле Брасс, Чандра Бхагаватула и Ежин Чой. Winogrande: Конкурирующая схема винограда в масштабе. Сообщения ACM, 64(9):99–106, 2022.
- [22] Мартен Сап, Ханна Рашикн, Дерек Чэн, Ронан Лебрас и Ежин Чой. Socialqa: Общественные рассуждения о социальных взаимодействиях. *arXiv preprint arxiv:1904.09728*, 2018.
- [23] Мирак Сузыун, Натан Скайлз, натанаэль Шерли, Себастьян Германн, И Тай, Хюн Вон Чун, Ааканкша Чаудхури, Куок В Ле, Эд Х Чи, Денни Чжоу, и Джейсон Вэй. Сложные задачи большого стола и может ли цепочка мыслей решить их. *arXiv preprint arxiv:2210.09261*, 2023.
- [24] Алон Талмор, Джонатан Херциг, Николас Лури и Джонат Берант. Commonsenseqa: Задача ответа на вопрос, направленная на знания общего смысла. *arXiv preprint arxiv:1811.00937*, 2017.
- [25] Юго Тувро, Тибо Лаврил, Готье Изакар, Ксавье Мартине, Мари-Анн Лашо, Тимоте Лакруа, Батист Розьер, Наман Гояль, Эрик Амбро, Фейсал Азхар и др. Llama: Open and efficient foundation language models. *arXiv preprint arxiv:2302.13971*, 2018 (неопр.) – Справочник по языкам программирования.
- [26] Хьюго Тувро, Луи Мартин, Кевин Стоун, Питер Альберт, Амджад Альмахейри, Ясмин Бабаэи, Николай Башлыков, Сумья Батра, Праджвал Бхаргава, Шрүти Бхосале и др. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arxiv:2307.09288*, 1998. (недоступная ссылка)
- [27] Ашиш Васвани, Ноам Шазир, Ники Пармар, Якоб Ушкорейт, Лион Джонс, Айдан Н Гомес, Лукаш Кайзер и Илья Полосухин. Внимание – все, что вам нужно. Прогресс в нейронных системах обработки информации, 30, 2017.
- [28] Роэн Зеллер, Ари Холтсман, Йонатан Биск, Али Фархади и Ежин Чой. Hellaswag: Может ли машина действительно закончить ваше предложение? *arXiv preprint arxiv:1905.07830*, 2018.
- [29] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: Человекоцентричный эталон для оценки моделей фундамента. *arXiv preprint arxiv:2304.06364*, 2022.