ВАДИМ ЖАРТУН

ГЛАВНОЕ ПРО AI

ЧТО ВАЖНО ЗНАТЬ ДЛЯ ВНЕДРЕНИЯ
AI В БИЗНЕСЕ

Вадим Жартун

Главное про AI

«Автор»
2026

Жартун В.

Главное про AI / В. Жартун — «Автор», 2026

Это не научный труд, не художественное произведение и не одна идея, растянутая на 100 страниц, а практическое руководство для тех, кто хочет разобраться в AI — и начать использовать его в работе уже завтра.Это книга для собственников, руководителей, офисных работников — для всех, кто видит в AI перспективу и не хочет полагаться ни на обещания маркетологов, ни на страшилки скептиков.Из неё вы узнаете:- где нейросети обходят человека, а где ему уступают;- когда выбирать облачные модели, когда — локальные, и не переплатить;- как ставить задачи модели, чтобы получать результат;- чем AI-агент отличается от чат-бота и когда он нужен;- какие процессы в продажах, маркетинге и управлении стоит отдавать AI;- как считать ROI от AI-внедрения;- как собрать личный AI-стек и внедрять AI в команде;- как обеспечить безопасность AI;- куда движется индустрия AI.Чек-листы, списки и таблицы помогут вам сориентироваться в теме AI и принять взвешенные решения и том, куда двигаться дальше.

Содержание

Вадим Жартун
Главное про AI

Посвящается моей любимой жене, моей Надежде.

Предисловие

Это не научный труд, не художественное произведение и не одна идея, растянутая на сотню страниц. Я написал эту книгу для тех, кто уже попробовал работать с AI и теперь хочет понять, что это такое и что делать дальше, чтобы эффективно использовать AI в бизнесе: для собственников, менеджеров, юристов, дизайнеров, финансистов, инженеров — не специалистов в IT или машинном обучении. Для тех, кто видит для себя в AI перспективу или угрозу.

Внедрение AI в компании, подразделении или на своём рабочем месте поднимает массу практических вопросов: можно ли ожидать от AI полезных результатов в конкретной области, какие инструменты и модели лучше выбрать, какой экономический эффект это даст, как внедрять AI-решения, какие подводные камни вас ожидают на этом пути, чего стоит ожидать от AI в ближайшем будущем, и так далее. Каждая из 17 глав книги — содержательный ответ на наиболее острые из этих вопросов.

Не стану врать, что тема AI проста и всё можно объяснить за пять минут на пальцах. Это не так, поэтому книга получилась довольно объёмной. Чек-листы, списки и таблицы помогут вам быстро сориентироваться в теме и принять взвешенные решения, основываясь не на радужных обещаниях маркетологов, страшилках вечных скептиков или эмоциональных высказываниях популярных блогеров, а на фактах.

В конце каждой главы есть короткое резюме с ключевыми мыслями, чтобы можно было быстро погрузиться в контекст в случае необходимости, а специфические термины и аббревиатуры, которые вы не обязаны знать, я постарался расшифровать не только в глоссарии, но и прямо в тексте.

Разумеется, эта книга про AI была написана с помощью AI. В основу её текстов легли материалы моих семинаров на стыке искусственного интеллекта и менеджмента, а также публикации в открытых источниках. На AI легла работа по переводу текстов, верификации данных, корректуре и множеству других технических задач. Для такой динамично развивающейся области знаний это просто необходимо, иначе книга рискует устареть ещё до момента публикации. Такова реальность, и мы не можем её игнорировать.

Спасибо, что открыли эту книгу. Давайте начнём.

Глава 1. Как устроены LLM

Летом 2017 года восемь инженеров Google Brain в Маунтин-Вью спорили, можно ли отказаться от рекуррентности в архитектуре нейросети и оставить только механизм внимания — научить машину смотреть на предложение целиком, а не читать слово за словом. Для тех, кто привык к рекуррентным сетям, затея звучала ересью. Базовую модель они обучили за двенадцать часов, большую — за три с половиной дня, тоже на восьми видеокартах; результаты англо-французского теста пришли за пять минут до дедлайна. Двенадцатого июня 2017 года на arXiv появилась статья «Attention Is All You Need» («Внимание — это всё, что вам нужно»), и через восемь лет у неё было больше ста семидесяти тысяч цитирований — это входит в топ-10 самых цитируемых научных работ двадцать первого века. Эта дата — точка отсчёта для всей современной индустрии AI (Artificial Intelligence, искусственный интеллект).

Я начинаю с этой истории, потому что она отвечает на вопрос, с которым вы пришли в книгу: «что я использую каждый день, когда ввожу запрос в чат?». Короткий ответ: статистическую машину, обученную на терабайтах текста предсказывать следующее слово по предыдущим. Длинный ответ — почему эта машина вообще смогла появиться. Три вещи сошлись в одной точке: большие корпуса текстов, видеокарты, способные их перемалывать, и архитектура, которая хорошо подходит для этих видеокарт. Дальше сработала обычная инженерная логика: что работает, то повторяем в большем масштабе.

АНАТОМИЯ ТРАНСФОРМЕРА: АРХИТЕКТУРА, ИЗМЕНИВШАЯ ВСЁ

Когда вы слышите в новостях «новая модель от вендора X», важно знать: в основе большинства моделей, с которыми вы работаете, лежит одна и та же архитектура — трансформер. Это архитектура нейросети, появившаяся в 2017 году (историю её создания я рассказал в начале главы). Трансформер лежит в основе современных LLM (Large Language Models, большие языковые модели) — и именно о них эта глава. Слова «трансформер» и «LLM» — не синонимы: LLM — это конкретная обученная модель, а трансформер — архитектура, по которой её собрали. Одну и ту же архитектуру можно обучить на разных данных и получить разные модели, поэтому на самом деле по этому чертежу собраны и Qwen, и DeepSeek, и GigaChat, и YandexGPT.

ТРАНСФОРМЕР БЕЗ ФОРМУЛ: СУТЬ НА ПАЛЬЦАХ

Никакой мистики. Трансформер смотрит на всё предложение целиком и для каждого слова решает, с какими другими словами его связать. До него текстовые нейросети — рекуррентные сети (RNN, recurrent neural networks) — читали предложения слово за словом, как вы читаете эту строку: чтобы понять десятое слово, нужно было сначала «переварить» девятое, а до него — восьмое, и так до первого. RNN по природе последовательны, и на видеокарте с тысячами ядер реально работало меньше десяти процентов — остальные ждали, пока одна цепочка токенов ползёт от первого к последнему. Трансформер убрал цепочку: каждое слово за один проход связывается со всеми остальными, и эту операцию можно делать параллельно на тысячах ядер. Та же загрузка видеокарты, что раньше не превышала десяти процентов, выросла до девяноста, и с 2017 года размер моделей стал расти на порядки — инженерная логика «что работает, то повторяем в большем масштабе» перестала упираться в железо. Суть трансформера простая: нейросеть, которая хорошо параллелизуется. Не «более умная» — лучше приспособленная к железу.

МЕХАНИЗМ SELF-ATTENTION

Почему трансформер «победил» RNN, что именно изменила архитектура внутреннего внимания (self-attention) и какие ограничения железа она сняла? Именно этот механизм реализует возможность каждому слову смотреть на все остальные в предложении. Работает он в три шага.

Сначала для каждого слова модель строит три вектора — запрос (Query), ключ (Key) и значение (Value). Это не три разных вычисления, а три проекции одного и того же слова — три угла зрения на одну сущность.

Затем она сравнивает запрос слова с ключами остальных и получает набор «весов внимания» — численную оценку того, насколько каждое слово важно для текущего. Веса нормализуются так, чтобы в сумме давать единицу, и по ним берётся взвешенная сумма значений остальных слов.

На выходе — новое представление слова, «обогащённое» контекстом.

Простой пример. В предложении «он взял ключ, потому что дверь была закрыта» слово «он» при обученном механизме получает высокий вес связи с тем существительным, к которому относится это местоимение, а «ключ» — с «дверь». RNN утрамбовывала весь смысл в одно скрытое состояние. Трансформер каждый раз заново смотрит на весь контекст.

Операция повторяется не один, а 8–16 раз параллельно с немного разными матрицами. Это называется многоголовым вниманием (по-английски — multi-head attention). Каждая «голова» учится решать свою задачу: одна ловит грамматические связи — какое слово является подлежащим, другая — смысловые: о каком предмете речь, третья — кто к кому относится в длинном предложении. Головы склеиваются — получается вектор, который пойдёт в следующий слой сети. Никакой магии: умножение матриц, выполняемое параллельно на тысячах ядер видеокарты.

ПОЧЕМУ ТРАНСФОРМЕР СТАЛ СТАНДАРТОМ

Трансформер победил не потому, что он единственный возможный, а потому что совпали три свойства, которые инженеры ценят больше красоты решения.

Масштабируемость через железо. Когда загрузка видеокарты поднялась с десятков процентов до почти полной, дальше работала обычная инженерная логика: что работает, то повторяем в большем масштабе. И оказалось: чем больше параметров и данных, тем предсказуемо лучше модель работает. Эту закономерность в 2020 году закрепили в OpenAI, а в 2022-м DeepMind уточнил рецепт (разберём подробно в разделе про то, почему модели появились именно сейчас).

Качество обобщения. Механизм внутреннего внимания работает не только на коротких предложениях, но и на длинных текстах, коде, таблицах — везде, где есть отношение «этот элемент зависит от того».

Универсальность. Один и тот же блок self-attention пригодился не только для текста: его используют для картинок, разбитых на куски, для звука, для белковых последовательностей. Подход не застрял в NLP — он стал общим языком современного AI.

Следующий раздел — о том, что именно модель предсказывает и как из цепочки предсказанных токенов складывается связный ответ.

ИГРА В УГАДЫВАНИЕ ТОКЕНА: ЧЕМ НА САМОМ ДЕЛЕ ЗАНЯТА МОДЕЛЬ

Представьте, что вы печатаете в чате «Премьер-министр Великобритании…». Модель выберет то слово, которое статистически чаще встречалось в похожем контексте в обучающих текстах. «Заявил», «встретился» или «подал в отставку» — не магия и не мышление, а статистика такого объёма, который ни один человек не способен охватить. Что такое токен, разберём в отдельном разделе ниже, пока запомните его как «фрагмент текста, на который модель разбивает входные данные». CSET, Центр безопасности и новых технологий Джорджтаунского университета, формулирует это так: «правильный входной запрос превращает машину для предсказания следующего слова в машину для ответа на вопросы». Именно это и происходит в каждом чате: вы формулируете задачу — модель продолжает текст так, как, по её подсчётам, должно следовать дальше в этом контексте.

Себастьян Рашка (*Sebastian Raschka*), автор книги «Собрать большую языковую модель с нуля», описывает механизм детальнее. Во время обучения модель обрабатывает последовательность токенов и на каждой позиции выдаёт вектор оценок для всего словаря. Вектор превращается в распределение вероятностей, и модель «штрафуется», когда приписывает низкую вероятность настоящему следующему токену. За миллионы пакетов данных и миллионы документов она учится не правилу «какое слово следующее», а тому, какие продолжения в каком контексте чаще встречаются. Постепенно усваивает синтаксис, смысл, стиль, факты и длинные языковые закономерности. «Выучить язык» в техническом смысле — это не «понять правила», а «выучить, какие слова в каких контекстах идут друг за другом».

Понимает ли модель то, что говорит, или только очень хорошо угадывает следующее слово? Модель не «понимает» намерение пользователя как человек и не знает, какой токен «правильный», — только какой вероятный в её статистике. Пять вещей, которые она не делает «сама по себе», важно знать заранее — иначе вы будете ждать от неё того, чего она не умеет:

1: **Не проверяет факты.**Модель продолжает текст по статистике, а не исходя из реальности.

2: **Не обращается к базам данных**без подключённого внешнего инструмента — её «знание о мире» заморожено на дате отсечки обучающего корпуса.

3: **Не возвращается назад**, чтобы отредактировать уже написанное, — каждый токен добавляется к предыдущим, всё сказанное остаётся как было.

4: **Не планирует ответ заранее**— она генерирует по одному токену, и каждый следующий токен зависит от уже написанного.

5: **Не останавливается «сама по себе»**— стоп-сигнал задаётся извне: по лимиту длины, по специальному токену или по команде пользователя.

Почему модель так уверенно врёт и при чём тут отсутствие внутреннего «детектора правды»? Понимание этого пригодится вам каждый раз, когда модель выдаст уверенный, но неправильный ответ. Модель не «забыла», не «ошиблась», не «решила вас обмануть»: она выбрала токен, который статистически подходил лучше всего, и ей было не на что опереться, кроме вероятностей. Уверенный тон — не признак точности. Чем выше вероятность следующего слова в обучающих текстах, тем увереннее модель его произносит.

В 2023 году в США на судебном процессе Mata v. Avianca выяснилось, что юрист Стивен Шварц (*Steven Schwartz*), использовавший ChatGPT для подготовки иска, подал в суд шесть несуществующих судебных прецедентов и подкрепил их цитатами, которых не было в источниках. Судья оштрафовал его. Этот случай показал то, что статистическая природа делает неизбежным: модель не отличает правду от правдоподобия и на любой запрос выдаст ответ, который звучит правильно, даже если за ним ничего не стоит. Это не баг, который исправят в следующей версии. Это системное свойство всех современных языковых моделей.

С этим утверждением спорят. Джеффри Хинтон (*Geoffrey Hinton*), лауреат Нобелевской премии 2024 года за работы в области нейросетей (формально — премия по физике, совместно с Джоном Хопфилдом (*John Hopfield*)), считает иначе: «чтобы точно предсказывать следующее слово, нужно понимать предложение» (оригинал: *To predict the next word accurately, you have to understand the sentence*). По Хинтону, при достаточном масштабе статистика порождает понимание как попутный эффект, и спор идёт не о том, статистическая ли модель, а о том, считать ли возникающие при масштабе способности пониманием.

Это контраргумент к метафоре «стохастического попугая» — так называют модель, которая воспроизводит услышанное, не понимая смысла; метафору в 2020 году ввели Тимнит Гебру (*Timnit Gebru*), Эмили Бендер (*Emily Bender*) вместе с коллегами. Все позиции сходятся в том, что модель статистическая, и разница только в том, считаем ли мы возникающие при масштабе способности пониманием. Спор философский — и не мешает использовать LLM как инструмент.

КОНТЕКСТНОЕ ОКНО: ГРАНИЦЫ И «ПОТЕРЯННАЯ СЕРЕДИНА»

У модели нет памяти в человеческом смысле. Есть контекстное окно — ограниченный «карман», в который она смотрит, продолжая текст. Каждый новый фрагмент она видит, держа в голове всё, что помещается, без потерь и сжатия. Окно кончилось — предыдущий текст из головы выпадает. Модель не помнит, что было за пределами окна, и не предупреждает об этом. Просто работает с тем, что поместилось.

Представьте, что вы читаете книгу через узкое окошко в двери. Страница за страницей ползут мимо, и в каждый момент вы видите только то, что в окошке помещается. Всё, что осталось за его пределами, для вас не существует — вы не помните, что было на предыдущих страницах, потому что их нет в поле зрения. Загрузили в чат пятидесятистраничный договор, а окно рассчитано на двадцать страниц, — модель физически увидит только последние двадцать. Начала договора для неё в этот момент просто нет. Переспрашивать бесполезно.

Контекстное окно важно для офисного работника по двум причинам. Для русского текста то же окно вмещает почти вдвое меньше слов, чем для английского, и платить за токены приходится в полтора-два раза больше. Окно измеряется в токенах, а не в словах, и до сих пор я говорил об этом как о факте — пора объяснить, что такое токен и почему он для русского обходится дороже. Это тема следующего раздела.

Модель физически работает не с текстом, а с числами. Каждое слово или кусок слова превращается в токен, и только после этого попадает в нейросеть. По оценке OpenAI (GPT-4 tokenizer release notes, 2023), сто токенов — это примерно семьдесят пять английских слов. В русском текст дробится на больше токенов, и то же окно в 128 000 вмещает не сто тысяч русских слов, а пятьдесят-шестьдесят тысяч. Когда маркетинг пишет «модель читает 100 000 слов», он имеет в виду английские слова, и для русского офисного текста это число надо делить примерно пополам.

Восемь тысяч токенов — это короткое письмо, протокол одного совещания, абзац кода. Тридцать две тысячи — статья средней длины, нормальный договор на двадцать-тридцать страниц, развёрнутый диалог на десяток ходов. Сто двадцать восемь тысяч — годовой отчёт небольшой компании, переписка за квартал, техническая книга. Миллион — стопка из десяти годовых отчётов или вся документация одного продукта. Для большинства офисных задач хватает 32–128 тысяч токенов, и покупать миллионное окно ради того, чтобы «уже точно хватило», — это как заказывать грузовик для поездки за хлебом. К середине 2026 года окно в миллион токенов стало стандартом у ведущих вендоров — GPT-5.5, Claude Opus 4.6+ и Gemini 3.1 Pro заявляют миллион, Llama 4 Scout — десять миллионов, Qwen3-Max Plus и DeepSeek V4 — по миллиону токенов. Но заявленный миллион и рабочий миллион — две разные вещи:

по независимым оценкам, реальная ёмкость стабильно держится на 60–70% от рекламного максимума, а середина всё так же «теряется». Так что в 2026 году для офисного работника формула выбора простая: для письма и протокола — 32K, для длинного документа — 128K, для миллиона — отдельный повод подумать, действительно ли вам нужен миллион или вы просто загружаете всё подряд.

Имеет значение, куда ставить ключевую инструкцию. Самый опасный момент в работе с длинным диалогом — не тот, когда модель отказывается отвечать, а тот, когда она соглашается. Когда вы выходите за пределы окна, платформа либо возвращает ошибку «превышена длина контекста», либо — и это случается чаще — молча обрезает начало разговора и продолжает отвечать так, будто ничего не произошло. В первом случае вы хотя бы видите сбой. Во втором модель отвечает на вопрос, опираясь на усечённый контекст, и вы уверены, что она «всё помнит». Вот это и опасно: забывание без явного сигнала — маскировка потери памяти.

В 2023 году исследователи из Стэнфорда и Принстона во главе с Нельсоном Лю (*Nelson Liu*) обнаружили ещё одну ловушку, которую они назвали «потерянной серединой» (*Lost in the Middle*). Длинный документ с важной деталью в центре модель обрабатывает хуже, чем короткий: она лучше помнит начало и конец контекста, хуже — середину. Загрузили в чат договор на девяносто страниц и просите «найди пункт про досрочное расторжение», а нужный пункт попал в середину, — модель с высокой вероятностью его пропустит, даже если формально весь документ в окне. Практический вывод: важное ставьте в начало и в конец, середину не загружайте без необходимости.

Как размер контекста влияет на выбор модели для работы с большими документами, разберёмся чуть позже. Сейчас перейдём к токенизации.

ТОКЕНИЗАЦИЯ: СКРЫТАЯ НАЦЕНКА НА РУССКУЮ РЕЧЬ

Модель работает не с буквами и не со словами, а с числами. Прежде чем показать ей ваш договор, текст придётся нарезать на кусочки — токены, — и каждому кусочку присвоить числовой идентификатор. Слово «договор» может стать одним токеном, а может развалиться на три: «дог», «ов», «ор». Как именно — решает токенизатор, не грамматика и не ваш редактор. Когда в диалоговом окне ChatGPT вы набираете запрос, происходит короткий ритуал, о котором вы и не подозреваете. Система берёт ваш текст, прогоняет его через токенизатор и получает на выходе последовательность чисел вроде 101 2054 2003 2651 1005 — условных идентификаторов из словаря модели, где каждое число соответствует своему кусочку текста. Эти числа уходят в нейросеть, нейросеть их перемалывает и выдаёт обратно тоже числа, а уже отдельный механизм превращает их в человеческие буквы на экране.

Алгоритм, который режет текст на токены, появился задолго до нейросетей. В 1994 году Филип Гейдж (*Philip Gage*) предложил метод побайтового кодирования пар — способ сжать текст, заменяя самые частые пары букв одним символом. В 2018 году инженеры OpenAI приспособили этот же приём под GPT, и с тех пор он работает внутри каждой большой языковой модели. Идея простая: берём текст, считаем, какие пары букв рядом встречаются чаще всего, склеиваем их в один символ, повторяем, пока не наберём словарь нужного размера. Этот базовый метод называется BPE — алгоритм нарезки на пары символов (Byte Pair Encoding). WordPiece и SentencePiece — варианты BPE с небольшими отличиями в выборе пар; для офисного работника разница между ними не важна, вы этот алгоритм не выбираете.

Почему не «по словам» и не «по буквам»? Потому что оба варианта — крайности, и обе плохо работают. Если резать по буквам, скромное «привет» превратится в шесть токенов, и модели придётся каждый раз заново собирать слово из букв. Если резать по словам, словарь раздуется до миллионов форм: «бежать», «убежал», «забежал», «перебежал», «выбежал» — каждое нужно хранить отдельно, и для редких слов просто не хватит места. Токенизатор наре-

зает текст на кусочки промежуточного размера: целые слова, если они частые, и куски слов, если слово редкое или длинное.

У токенизатора есть фиксированный словарь — набор кусочков, которые он умеет распознавать. Словарь составляется один раз, на этапе обучения, и в нём столько русских кусочков, сколько их было в обучающих текстах. По данным OpenAI (GPT-3.5/4 tokenizer release notes), у GPT-3.5 и GPT-4 словарь — около ста тысяч кусочков, из них кириллических меньше пятисот. У GPT-4o словарь расширили до двухсот тысяч, и кириллических стало около четырёх с половиной тысяч — в десять раз больше (по анонсу OpenAI 2024 года). В OpenAI осознали проблему и вложились в неё деньгами, но и четыре с половиной тысячи на двухсоттысячный словарь — это всё ещё два процента, а не половина.

Чтобы почувствовать разницу на цифрах, возьмём простую фразу «Я встретил огромную собаку» и её английский перевод «I met a huge dog». Английский вариант превращается примерно в 5–6 токенов, русский — в 12–14. У моделей с латинским алфавитом в словаре больше целых английских слов, и они реже режут текст на части; у кириллических слов словарь беднее, и токенизатор рубит слово на куски. Точные цифры зависят от модели, но порядок сохраняется у большинства современных LLM.

Что это означает для офисного работника на практике:

1: **Скорость ответа падает.**Модель генерирует токены последовательно, и чем их больше, тем дольше идёт ответ.

2: **Качество падает на сложной лексике.**Юридические, медицинские и финансовые термины русского языка часто режутся на нерегулярные куски, и модель хуже «понимает», что перед ней.

Вопросы цены и размера окна для русского текста мы разобрали в предыдущем разделе, поэтому здесь остановимся только на этих двух следствиях.

Зачем нужны модели с поддержкой кириллицы на уровне токенизатора и что это даёт на практике?

GigaChat, YandexGPT, Qwen, DeepSeek появились с одной и той же мыслью: словарь нужно наполнить кусками тех языков, на которых модели будут работать, и не наполнять его кусками тех, на которых не будут. Для русскоязычной задачи кириллическая модель часто оказывается дешевле и быстрее западной, и это не «преимущество российской инженерии», а математический факт: у неё больше кириллических токенов в словаре, и она меньше тратит их на одну и ту же фразу. К 2026 году выбор стал конкретнее: YandexGPT 5 Pro встроен в Алису и закрывает офисные задачи на русском в формате подписки — платите фиксированно, без отдельной оплаты за токены, GigaChat 2 Max доступен в варианте локальной установки (on-prem, на серверах компании) под ФЗ-152 (Федеральный закон «О персональных данных»), а Llama 4 (выпущенная в апреле 2025 года) русского в списке языков дообучения не содержит — и для русскоязычной задачи это явный сигнал смотреть в сторону кириллических моделей. Миф «кириллица работает хуже, потому что модель обучена на английском» — неверен: проблема в токенизации, не в качестве обучения, и решается она не переходом на западную, которая выглядит «лучше» в маркетинговых таблицах, а выбором кириллической модели с собственным словарём.

АВТОРЕГРЕССИЯ, ДИФФУЗИЯ, ГИБРИДЫ: ТРИ СЕМЕЙСТВА МОДЕЛЕЙ

Внутри категории «большая языковая модель» работают три семейства, и по названию сервиса уже можно прикинуть, чего от него ждать. Для офисного текстового чата вам нужны

авторегрессивные модели — GPT, Claude, Gemini, GigaChat и YandexGPT; остальные два семейства для чистого текста почти не используются.

Чтобы сравнить три семейства бок о бок, соберём ключевые признаки в таблицу.

Таблица 1.1. Три семейства языковых моделей — назначение и скорость ответа

Семейство	Что делает	Типичные задачи	Скорость ответа
Авторегрессивные	Генерирует токен за токеном слева направо	Текст, код, диалог	Секунды–минуты
Диффузионные	Убирает шум из хаоса за несколько проходов	Картинки, видео, новая текстовая ниша	Минуты для картинки, часы для видео
Гибридные (мультимодальные)	Склейка: авторегрессия для текста + диффузия для картинок	Текст + картинка + голос в одном окне	Зависит от части

Таблица показывает разницу в скорости и задачах: авторегрессия — секунды-минуты для текста, диффузия — минуты для картинки и часы для видео, гибрид — зависит от того, какую часть задействуете. Для чисто текстовой работы гибрид избыточен и часто медленнее, а диффузионная модель для текста в 2026 году всё ещё экзотика.

Авторегрессия объясняет три вещи, которые вы видите каждый день в чате:

1: **Потоковый вывод.**Ответ появляется токен за токеном, и вы видите, как модель «печатает» в реальном времени.

2: **Невозможность «отмотать назад».**Модель не может вернуться к началу ответа и поправить его — каждый токен добавляется к предыдущим.

3: **Цепочка рассуждений (chain-of-thought).**На длинных задачах модель использует «думание вслух» — каждый шаг рассуждения опирается на предыдущий.

Диффузионные модели решают принципиально другую задачу. Они не «пишут» текст или картинку, а учатся убирать шум из хаоса. Это та же механика, по которой проявляется фотография в тёмной комнате: берётся поле случайных пикселей, и за двадцать-тридцать аккуратных проходов из шума «проступает» изображение. Диффузия плохо работала с текстом до 2025 года, но почти все генераторы картинок — Midjourney, DALL-E 3, Stable Diffusion 3, Kandinsky 5.0 — это диффузионные модели. В 2025 году появились первые коммерческие диффузионные LLM — Mercury от Inception Labs (калифорнийский стартап, специализирующийся на диффузионных языковых моделях), которые выдают текст в 5–10 раз быстрее авторегрессии. Для офисного работника в 2026 году это всё ещё экзотика, а не рабочий инструмент. Полезно знать о них не ради выбора, а ради того, чтобы понимать, почему картинки в чате рисуются почти мгновенно, а текст — посимвольно: у них физически разная архитектура.

Гибридные (мультимодальные) модели — для офисного работника это самый заметный сдвиг последних двух лет. GPT-4o, Gemini 2.x, Claude 4, Llama 4 — все они в одном и том же окне принимают и выдают текст, картинки, аудио и видео. Архитектурно это «склейка»: авторегрессионная часть генерирует токены изображения, диффузионный декодер превращает их в пиксели. Для офисного работника теперь не нужно формулировать «запрос для художника» отдельно, можно просто продолжить разговор. Когда сервис принимает картинку в сообщении

и описывает, что на ней, — это распознавание картинки на входе (по-английски — vision-in). Когда сервис рисует картинку прямо в чате без отдельного сервиса — это генерация картинки на выходе (по-английски — image-out), и это почти всегда гибрид.

По названию сервиса можно прикинуть, к какому типу он относится: текстовый чат с рассуждениями — авторегрессия, генератор картинок — диффузия, «всё в одном» — гибрид. Для смешанных задач (объяснить схему, перевести вывеску, сделать презентацию) гибрид единственный, кто работает без склеек.

Имея в голове эти три семейства, посмотрим, почему они вообще появились и смогли развиться до рабочих инструментов за последние годы.

ПОЧЕМУ ВСЁ ЭТО СЛУЧИЛОСЬ ИМЕННО СЕЙЧАС

Современная языковая модель — не одно изобретение и не одно озарение, а совпадение трёх факторов, каждый из которых сам по себе недостаточен. Убери любой из трёх — и большие языковые модели либо не появляются, либо приходят на десять лет позже. Это и есть ответ на вопрос «почему именно сейчас»: впервые в истории у одного поколения инженеров оказались все три инструмента в одной связке.

Эти три фактора подтверждаются тремя вехами:

– **Данные.**В 2009 году Фей-Фей Ли (*Fei-Fei Li*) из Стэнфорда с командой разметили четырнадцать миллионов изображений через Amazon Mechanical Turk — этот датасет, ImageNet, стал первым датасетом такого масштаба. Через десять лет тот же рецепт перенесли на текст: Common Crawl (некоммерческий проект, который с 2008 года архивирует веб-страницы) выкачивает весь интернет, GPT-3 берёт оттуда восемьдесят два процента своих токенов.

– **Железо.**В 2012 году аспирант Алекс Крыжевский (*Alex Krizhevsky*) из Торонто обучил нейросеть AlexNet на двух игровых видеокартах NVIDIA GTX 580, написанных на CUDA (язык, который NVIDIA выпустила в 2007 году, чтобы видеокарты перестали быть «только для игр»), и AlexNet выиграл конкурс ImageNet с отрывом в десять с лишним процентных пунктов. Стало ясно: видеокарты — не побочный продукт игровой индустрии, а отдельный инженерный слой.

– **Алгоритм.**Трансформер 2017 года дал параллелизацию, и за пять лет размер моделей вырос на три порядка.

В январе 2020 года OpenAI публикует статью о масштабировании моделей (Kaplan и соавторы, «Scaling Laws for Neural Language Models» — «Законы масштабирования для нейросетевых языковых моделей»), которая задала стратегию обучения на годы вперёд. Тезис простой: чем больше модель и чем больше данных, тем предсказуемо лучше она работает, причём зависимость степенная. В марте 2022 года DeepMind показывает, что OpenAI ошибся в одном: их рецепт недооценивал данные. Модели, обученные «маленькими, но долго», били модели, обученные «большими, но быстро». Так появилось эмпирическое правило из работы DeepMind под названием Chinchilla (это именно правило обучения, а не модель — название отсылает к шиншилле из детской повести): на каждый параметр модели нужно примерно двадцать токенов данных. То, что раньше делали интуитивно, стало инженерной дисциплиной.

Хинтон не изобрёл нейросети в 2010-х — он работает с ними с 1980-х, и большая часть его карьеры пришлась на период, когда эту область называли безнадёжной. Авторы «Attention Is All You Need» — восемь человек из Google Brain, а не один гений с армией ассистентов. LLaMA — десятки инженеров Meta и результат трёх лет подготовительной работы. Современная большая модель — это индустриальный продукт, где без инфраструктуры не будет про-

рыва. «История AI как история одиноких гениев» — пиар-история, которая хорошо продаётся, но плохо отражает реальность.

Миф «модель знает всё, потому что обучена на всём интернете» — неверен. Она обучена на срезе с пропусками и с датой отсечки. У неё нет представления о «сейчас», есть только статистика по корпусу, который заканчивается в определённый месяц, — отсюда уверенно поданный старый прецедент как свежий. То, что вы используете, делала команда из сотни человек с бюджетом в десятки миллионов долларов, а не один талантливый исследователь.

Какие сегодняшние «фундаментальные» ограничения AI через два-три года перестанут существовать и что, наоборот, подорожает? Узкое место в индустрии сдвигалось три раза, и каждое десятилетие приносило свой ответ:

– **2017 — алгоритм.** Трансформер дал параллелизацию.

– **2020 — данные и формула.** Chinchilla превратила «побольше данных» в инженерный план.

– **2024–2025 — применение модели и качество данных.** Узкое место сместилось снова.

Сейчас это не «как обучить» (железо есть) и не «на чём обучать» (текстов на английском хватает), а «как обработать миллион токенов контекста без квадратичного взрыва» и «как заставить модель думать дольше, не делая её работу слишком медленной». Индустрия отвечает тремя путями.

Разреженное внимание. Модель смотрит не на все токены, а на отобранные. Так работают DeepSeek и Gemini Flash.

Смесь экспертов (MoE — Mixture of Experts). Модель состоит из нескольких специализированных подмоделей, и для каждого токена выбирается свой эксперт.

Дополнительные вычисления на этапе ответа в режиме рассуждения. Так работают o1, Claude thinking и Qwen3-Max-Thinking.

Из этого следует: через два-три года текущие «фундаментальные» ограничения — длина надёжного контекста, качество рассуждений и цена топовых моделей — тоже перестанут быть узким местом. Это не «AI станет умнее», а «инженеры найдут способ тратить вычисления умнее». А подорожает то, что и всегда: электричество, видеокарты, инженерные кадры и качественные данные на русском.

ПАРАМЕТРЫ И МАСШТАБ: КОГДА РАЗМЕР ВАЖЕН, А КОГДА НЕТ

Когда в команде говорят «давайте возьмём модель побольше, чтобы точно работало», полезно иметь в голове список из трёх строк:

1: **7 млрд параметров — «карманный помощник».** Поправит письмо, сделает выжимку из протокола, переведёт короткий кусок, вытащит из длинного документа нужный пункт.

2: **70 млрд — «серьёзный собеседник».** Пишет код с пониманием структуры, разбирает юридические документы на десятой странице без потери нити, выдерживает сложную многошаговую инструкцию.

3: **405 млрд — «старший эксперт, которого вызывают по особому поводу».** Глубокий анализ, синтез противоречивых источников, длинные рассуждения с цепочкой выводов.

По моей оценке, около восьмидесяти процентов офисных задач закрывает 7-миллиардная модель, ещё пятнадцать — 70-миллиардная, и только оставшиеся пять действительно требуют 405-миллиардной. «Самая большая» для большинства команд не нужна.

Meta, выпуская Llama 3.1 в трёх размерах, фактически предлагает один и тот же рецепт, испечённый в трёх противнях: один для ноутбука, один для серверной стойки, один для датацентра. Цифра параметров отвечает на вопрос «сколько железа нужно», а не «насколько модель умна». По данным Meta, 405B обучена на более чем пятнадцати триллионах токенов на кластере из шестнадцати тысяч H100 — Meta подчёркивает, что 405B — первая модель Llama, обученная в таком масштабе. За числом стоит бюджет, а бюджет — не магия, а деньги, электричество и инженерные решения.

Microsoft в релизе Phi-4 (декабрь 2024) поставила курс на «больше — не всегда лучше»: модель с четырнадцатью миллиардами параметров позиционируется как рассуждающая, конкурирующая с куда большими моделями. В декабре 2025 Microsoft подвела итог: Phi-4-reasoning «соперничает с куда большими моделями на сложных задачах на рассуждение». В той же точке сходятся независимые исследования: стэнфордская работа о «мираже» эмерджентности (иллюзии резких скачков в способностях модели при увеличении размера; авторы — Schaeffer, Miranda, Koyejo) показала, что «резкие скачки» в способностях больших моделей могут быть артефактом того, как их измеряют. При смене метрики или при увеличении числа примеров скачки исчезают, и зависимость становится гладкой. «Больше параметров» — в первую очередь маркетинговый аргумент. Для большинства бизнес-задач хватает меньшей модели, а где 405B действительно нужна, вы это узнаете по характеру работы, а не по рекламе вендора.

Теперь, когда мы разобрали, что у модели под капотом, какие семейства бывают и как соотносить размер с задачей, посмотрим ретроспективно: почему всё это вообще появилось именно в последние годы, а не десятью раньше или десятью позже.

РЕЗЮМЕ

Модель в чате — статистическая машина, предсказывающая следующий токен по предыдущим. По этому чертежу собраны и GPT, и GigaChat, и YandexGPT, и Qwen. Разница между ними — в обучающих данных, не в инженерной идее. Уверенный тон модели — особенность распределения токенов, и в ближайших релизах это не изменится. Контекстное окно измеряется в токенах: для русского текста оно вмещает вдвое меньше слов, чем для английского. По названию сервиса вы прикидываете архитектуру: авторегрессия для текста, диффузия для картинок, гибрид для смешанных задач. LLM — индустриальный продукт, не изобретение одного исследователя. Для офисной рутины размер модели решает меньше, чем принято думать: основная масса задач закрывается 7-миллиардной моделью, и локальная 7B на ноутбуке часто выигрывает у облачной.

Утро понедельника. Вы открываете почту, там двадцать три письма, и у трёх из них вложения по пятнадцать страниц. До обеда надо ответить на семь, остальные — после. В пятницу босс попросил «сделать красиво» квартальный отчёт. Секретарь уволилась месяц назад, новую ещё не нашли. Звучит знакомо? Следующая глава — про то, какие из этих задач AI закроет за вас, какие возьмёт на половину, а какие лучше не подпускать к деловой переписке вообще. Это не фантастика и не утро офиса 2030 года — это типичная рабочая неделя 2027-го.

Глава 2. Что AI умеет и чем ограничен

Представьте конец 2025 года: двое менеджеров из разных компаний спорят за кофе. Один говорит: «AI отлично работает, я за день закрыл месячный отчёт». Второй: «А я попросил его написать досудебную претензию — и он сослался на статью закона, которой не существует. Хорошо, что помощник успел проверить». Оба говорят про одну и ту же модель, но видят её по-разному. Оба правы, и оба смотрят на одну и ту же машину. Разница — в типе задачи. Первый дал модели задачу с низкой ценой ошибки, черновик, — и сэкономил день. Второй дал задачу с высокой ценой, юридический документ, — сэкономил час, а потом рисковал карьерой. В следующих разделах разберёмся, как заранее отличить первое от второго — и перестать спорить с коллегами о том, «хорошо ли работает AI». Он работает по-разному. Это не каприз, а системное свойство.

ПЯТЬ КЛАССОВ ЗАДАЧ AI

Когда вы открываете чат и формулируете запрос, модель решает одну из пяти задач. Названия звучат сухо, но различение между ними — первая линия обороны против плохого результата. Важнее, чем выбор модели и длина промпта.

Классификация. Вы даёте модели текст и список меток, она возвращает метку. «Отнеси это письмо к одному из пяти типов: жалоба, запрос документов, претензия по качеству, повторное обращение, спам» — модель должна выбрать один из пяти вариантов:

1: жалоба;
2: запрос документов;
3: претензия по качеству;
4: повторное обращение;
5: спам.

Внутри та же машина, что рисует картины, но задача формализована настолько жёстко, что вероятность выдумки низкая. Хорошо работает: сортировка входящих, тегирование тикетов поддержки, фильтрация резюме по базовым критериям.

Извлечение фактов. Вы даёте договор и список полей, модель возвращает пары «поле — значение». «Из этого договора вытащи: дату, номер, стороны, предмет, сумму, срок, условия расторжения». Здесь модель работает как очень внимательный, но не очень надёжный стажёр. С задачей справляется, но если в договоре поля нет, она часто выдумывает значение вместо того, чтобы честно сказать «не нашла». Это системная особенность, и с ней борются отдельно — мы к этому вернёмся в разделе про распознавание галлюцинаций.

Резюмирование. Вы даёте длинный текст, модель возвращает короткий. Самая надёжная операция из всех. По данным Vectara Hallucination Leaderboard за ноябрь 2025, лучшие модели на задачах суммаризации выдают менее одного процента галлюцинаций — Gemini 2.0 Flash, например, показывает 0,7% (лучший результат рейтинга на тот момент).

Но разброс в рейтинге — от 0,7% до 20,2%. Выбор конкретной модели даёт двадцатикратную разницу в вероятности выдумки. «Лучшая модель» — не общий ярлык, а привязка к конкретной строке рейтинга. Для офисной рутины «сожми мне протокол в пять строк» это рабочий инструмент, и риск выдумки в этой задаче один из самых низких во всём, что умеет LLM.

Причина простая: при суммаризации модель работает с заранее данным текстом, и единственный способ выдумать — исказить исходник. Это случается редко, потому что текст у неё «перед глазами».

Генерация.Вы даёте тему, модель возвращает связный текст. «Напиши письмо клиенту о переносе встречи», «Составь должностную инструкцию для помощника», «Сформулируй пять вариантов слогана». Здесь качество сильно зависит от того, насколько точно вы описали требования: тон, формат, длину, аудиторию, обязательные элементы. Чем расплывчатее промпт, тем выше шанс получить красивый, но бесполезный для вашей задачи текст. Не потому что она «глупая», а потому что у неё нет вашего контекста.

Поиск и вопрос-ответ.Вы даёте вопрос, модель возвращает ответ. Самая опасная задача, потому что в ней модель работает без ваших документов, на своей внутренней статистике. «Кто был премьер-министр Японии в 2018 году?», «Какая статья ГК РФ регулирует расторжение договора аренды?», «Что говорит наш регламент о сроках согласования?» Последний вопрос — самый коварный: модель не знает вашего регламента, и её ответ будет выглядеть как «правда про регламент», но это выдумка в форме ответа. Про распознавание таких моментов — отдельный раздел ниже.

РАСЧЁТЫ: ЛОВУШКА КОМАНДЫ «ПОСЧИТАЙ МНЕ»

К пяти классам задач стоит добавить шестой — расчёты, потому что в нём модель ведёт себя иначе. Правило раздела такое: «посчитай мне» — плохой промпт, а «напиши Python-скрипт, который считает…» — хороший. Если в вашем продукте нет Code Interpreter, а задача про точную арифметику (НДС, проценты по кредиту, штрафы, курсовые разницы) — считайте калькулятором или скриптом, а не моделью.

Почему так. Когда вы просите модель посчитать, она не считает — она предсказывает, какой токен статистически вероятнее всего окажется следующим в последовательности «два плюс два». На простом сложении двузначных чисел модели сбиваются в 5–15% случаев, а на умножении трёхзначных — почти всегда. Исследователи Apple Machine Learning в работе GSM-Symbolic (октябрь 2024) показали это эмпирически: при изменении только числовых значений в задаче точность моделей колеблется в пределах 65 процентных пунктов, а добавление одного нейтрального «шумящего» предложения в условие обрушивает результат.

Современные AI-ассистенты с Code Interpreter решают эту задачу иначе. Модель не считает сама — она пишет скрипт, а Python в изолированной среде (sandbox) его исполняет. Из такой среды модель не имеет выхода в вашу файловую систему, интернет или чужие процессы. Результат вычисления возвращается модели как контекст, и она формулирует ответ на естественном языке. Сейчас в эту архитектуру умеют: OpenAI Advanced Data Analyses (бывший Code Interpreter), Anthropic Claude с калькулятором, Google Gemini Code Execution, Qwen Code Interpreter, Microsoft Copilot + Python в Excel.

На вопрос «сколько будет 2341 × 5678» без инструмента модель с вероятностью 70–80% ошибётся. С инструментом — выдаст точный результат.

Смешение типов в одном запросе — почти гарантированный путь к плохому результату. Когда вы пишете «прочти этот PDF, выдели риски, сравни с прошлогодним, перепиши как письмо директору и порекомендуй решение», вы заставляете модель одновременно извлекать факты, классифицировать риски, сравнивать, генерировать и формулировать рекомендацию. Модель делает один из пяти шагов прилично, остальные — кое-как, и вы получаете текст, в котором половина фактов выдумана, а половина — точная, но не к месту. Если задача сложная и многошаговая, разбейте её на последовательность простых запросов: сначала выжимка, потом — сравнение, потом — черновик письма. Пять минут на разбивку экономят час на правку.

Если вы ловите модель на выдумке и думаете «может, попробовать более новую версию», остановитесь. Версия тут ни при чём — дело в архитектуре.

RAG: ИНСТРУМЕНТ ПОДСТРАХОВКИ, А НЕ ЛЕКАРСТВО ОТ ГАЛЛЮЦИНАЦИЙ

Извлечение фактов и поиск — два самых опасных класса задач: в них модель работает без ваших документов и «додумывает» ответ. По данным Stanford HAI (2024), в этих классах модель выдумывает на 17–33% чаще, чем хотелось бы. Есть ли способ заставить её отвечать по вашим документам, а не по памяти? Есть, и он называется RAG.

В 2024 году индустрия воодушевилась идеей: если дать модели читать ваши документы, а не свои внутренние веса, она перестанет выдумывать. Технология называется RAG — Retrieval-Augmented Generation, или «поиск с подсказкой». На каждый запрос модель сначала ищет релевантные фрагменты в вашей базе знаний, а уже потом формулирует ответ на их основе. Идея элегантная, и для части задач она работает. Для другой части — нет, и важно понимать где.

RAG радикально снижает галлюцинации, когда вопрос имеет точный ответ в ваших документах. «Какой срок оплаты по договору с ООО „Ромашка”?», «Что говорит наш регламент о командировках?», «Какая версия политики безопасности действует с 1 января?» Без RAG модель отвечает из своей статистики и врёт уверенно. С RAG — опирается на конкретный фрагмент, и шанс выдумки резко падает.

Но RAG не серебряная пуля. В сентябре 2024 года команда Стэнфордского института человекоцентричного AI (HAI) и RegLab опубликовала исследование, которое отрезвило индустрию. Авторы — шесть исследователей во главе с Сугуном Магешем (*Sugun Magesh*) и известным NLP-учёным Кристофером Мэннингом (*Christopher Manning*) из Стэнфорда — взяли три коммерческих юридических AI-инструмента: Lexis+ AI, Westlaw AI-Assisted Research (построенный на GPT-4 после покупки Casetext за 650 миллионов долларов) и Ask Practical Law AI.

Цифры по галлюцинациям: Lexis+ AI — 17%, Westlaw — 33%, Ask Practical Law AI — 17%, базовый GPT-4 без RAG — 43%. RAG снизил долю ошибок примерно вдвое, но не устранил: в каждом шестом ответе коммерческие инструменты с доступом к проверенной юридической базе выдавали ложную информацию. Westlaw, построенный специально для юристов с фокусом на точность, галлюцинировал в каждом третьем ответе.

Практический вывод: в задачах с высокой ценой ошибки RAG — фильтр первого уровня, а не финальный контроль качества.

Миф «RAG решает проблему галлюцинаций» не выдерживает проверки: модель всё равно может переврать прочитанное или сделать вывод, которого в источнике не было. Грег Ламберт (*Greg Lambert*), главный специалист по знаниям в Jackson Walker, в комментарии изданию Artificial Lawyer по итогам Stanford-исследования сформулировал ключевую проблему — «creativity problem», «проблему творческого додумывания». Модель не просто ищет в базе, она «креативит» на основе найденного. Берёт реальные документы, соединяет их в несуществующую конструкцию и формулирует уверенный ответ со ссылками на «реальные» источники. Юрист видит красивый ответ со ссылками и не перепроверяет каждую строку.

В феврале 2024 года канадский Civil Resolution Tribunal (Британская Колумбия) признал авиакомпанию Air Canada виновной в неосмотрительном введении в заблуждение из-за того, что её чат-бот пообещал клиенту несуществующую скидку. Клиент Джейк Моффатт (*Jake Moffatt*), потерявший бабушку, спросил у чат-бота, можно ли оформить обратный билет со скидкой после похорон. Чат-бот ответил утвердительно и привёл подробный тариф. Air Canada использовала RAG-подобный поиск по собственной базе тарифов, но поиск вернул не тот документ, и модель сгенерировала уверенный ответ про несуществующую политику. На суде ком-

пания пыталась доказать, что чат-бот — «отдельная сущность, за которую она не отвечает». Трибунал отверг этот аргумент и обязал Air Canada выплатить Moffatt компенсацию в размере 812,02 канадского доллара (CAD). Кейс стал прецедентом: компания отвечает за то, что говорит её AI, даже если AI ошибся.

Вывод: RAG нужен в любом офисном проекте с собственными документами — без него модель выдумывает ответы «как будто из вашего регламента» из общей статистики. Но одного RAG недостаточно: нужно RAG с цитированием, чтобы каждое утверждение модели можно было проверить по конкретному фрагменту базы. И держать в голове: даже хороший RAG уменьшает галлюцинации, а не устраняет.

ТРИ СПОСОБА АДАПТИРОВАТЬ МОДЕЛЬ ПОД СЕБЯ

RAG снижает галлюцинации, но не устраняет их — мы это установили. Тогда возникает следующий вопрос: есть ли способ переделать модель под себя, чтобы она отвечала «в нашем тоне», «с нашей терминологией», «по нашим правилам»? На бытовом уровне это звучит как просьба «натренировать наш ChatGPT на наших документах». В техническом смысле под этой фразой скрываются три разные операции с разной ценой и применимостью, и путаница между ними стоит компаниям десятков тысяч долларов и месяцев работы впустую.

Промпт-инжиниринг с few-shot примерами (few-shot — обучение на нескольких примерах в самом промпте).Вы не трогаете модель вообще, а пишете ей в системной инструкции, как себя вести, и прикладываете 2–5 примеров «вопрос-ответ» в самом промпте. Это самая дешёвая операция: ноль рублей на технику сверх обычной стоимости API, плюс несколько часов работы сотрудника, который учится формулировать задачи. Подходит, когда задача помещается в шаблон, не требует своей базы знаний и сводится к управлению тоном, форматом, терминологией. Лимит: если few-shot в промпте «съезжает» с формата на длинных ответах, это сигнал, что задача вышла за границы инструкции.

RAG (поиск с подсказкой).Вы оставляете модель общую, но подключаете к ней свою базу знаний. При каждом запросе модель сначала находит в базе релевантные фрагменты, а уже потом отвечает с опорой на них. Стоимость — от единиц тысяч долларов на старте (векторная база, индексация, интеграция) до десятков тысяч на зрелом проекте. Подходит, когда у вас есть много собственных документов, на которые модель должна опираться, и эти документы регулярно обновляются. Преимущество RAG перед дообучением — свежесть: данные можно обновить за часы (переиндексировал базу), а не за недели (новый цикл обучения). Преимущество RAG перед промптом — цитируемость: сотрудник или клиент может ткнуть пальцем в конкретный пункт регламента, на который модель сослалась. В задачах с регуляторной ответственностью (комплаенс, медицина, финансы) RAG даёт прослеживаемость источника, и в ЕС по AI Act это юридически значимое преимущество, а дообученная модель — чёрный ящик, и доказать, на каком основании она выдала ответ, потом будет нечем.

Дообучение (fine-tuning).Вы берёте готовую модель и прогоняете её через дополнительное обучение на своих парах «вопрос-ответ», подкручивая её внутренние веса. Это самая дорогая операция. Полное дообучение модели 7B требует 100–120 ГБ видеопамяти — это аренда нескольких серверных NVIDIA H100 примерно за 50 000 долларов за цикл.

Стоимость дообучения GPT-4o у OpenAI в августе 2024 — 25 долларов за 1 миллион токенов тренировки плюс 3,75 за миллион токенов инференса дообученной модели (тарифы OpenAI на 2024 год, в 2026 могли измениться). По оценке Open Source Data Summit 2025, дообучение обходится в 10–50 тысяч долларов за итерацию и требует месяцев на подготовку данных, тогда как RAG-система разворачивается за недели при 10% от этой стоимости.

Подходит, когда у задачи специфический стиль или формат ответа, который не вытягивается через примеры в промпте: юридические заключения в формате конкретного арбитраж-

ного суда, медицинские протоколы в формате конкретной клиники, финансовая отчётность в формате конкретного банка, или стабильный классификатор на 12 типов риска, обученный на 5 000 размеченных договоров.

Промежуточный вариант — QLoRA. Это метод на основе LoRA (Low-Rank Adaptation), где к стандартному дообучению добавляется 4-битное квантование весов: веса модели хранятся не в полной точности, а в упрощённой, что и даёт экономию памяти. Для модели 8B требования к VRAM сокращаются с 69 ГБ до 16 ГБ, и дообучение укладывается в одну потребительскую видеокарту NVIDIA RTX 4090 за полторы тысячи долларов вместо пятидесяти. Но даже QLoRA остаётся дообучением со всеми его ограничениями: модель застывает на момент обучения, и при обновлении данных придётся запускать новый цикл.

Какой инструмент выбирать в какой ситуации. Промпт — для тона, формата и типовых шаблонов, где не нужны ваши документы. RAG — для всего, где есть собственная база знаний, которую нужно уважать (регламенты, договоры, политика). Дообучение — для устойчивого специфического стиля или классификатора, который не вытягивается через примеры. В большинстве офисных задач достаточно RAG с хорошей базой и грамотного промпта.

РАСПОЗНАТЬ ГАЛЛЮЦИНАЦИЮ: ГРАНИЦА ДОВЕРИЯ К ОТВЕТУ МОДЕЛИ

Способа, который даёт стопроцентную гарантию, нет, но есть пять маркеров, по которым выдумку ловит даже непрофессионал. Они не дают абсолютной уверенности, но сдвигают вероятность в вашу сторону.

1: **Подозрительная точность конкретных цифр.** Когда модель выдаёт «исследование Университета X от 2019 года показало, что 73,4% респондентов…», цифры выглядят слишком «гладкими» для реального исследования — повод проверить. Реальная социология даёт неровные числа вроде «71,3% опрошенных из выборки 1247 человек», потому что работает с погрешностями и доверительными интервалами.

2: **Неназванный источник.** «Установлено, что…», «По мнению экспертов…», «Как известно…» — типичные обёртки для выдумки. Нет конкретной ссылки на автора, работу или базу данных — факт вызывает сомнение по умолчанию.

3: **Длинный ответ с потерянной серединой.** Модель хорошо помнит начало и конец своего ответа и хуже — середину (потерянная середина, о которой мы говорили в главе 1). Если первые два абзаца и последний связаны по смыслу, а середина «плывёт» — там может быть выдумка.

4: **Внутренние противоречия при перефразировании.** Попросите модель «расскажи то же самое другими словами». Если при пересказе появились новые факты, которых не было в первой версии, первая версия была частично выдумкой, и пересказ её «достроил».

5: **Отсутствие оговорки про устаревание.** Модель уверенно пишет про события вчерашнего дня без оговорки «на момент последнего обновления моих данных» — это техническая выдумка: у модели нет данных «вчерашнего дня», и она не предупредила об этом.

Эти пять маркеров не равны «доказано, что модель врёт», но задают простую рабочую норму: если хотя бы один из пяти сработал — перепроверьте по первоисточнику, прежде чем использовать ответ в работе. Что делать, когда выдумка уже прошла и попала к клиенту, — следующий раздел.

AI ЛЖЁТ КЛИЕНТУ: МАСШТАБ УЩЕРБА И ЦЕНА ОШИБКИ

Пять маркеров из предыдущего раздела помогают ловить выдумку. Но что делать, когда выдумка уже прошла и попала к клиенту — в договор, претензию, иск? Mata v. Avianca, флоридские кейсы и Air Canada — это три примера того, как AI-враньё попало в клиентские документы. Здесь нужен следующий уровень защиты: разделение ролей и процесс контроля.

Юридический риск несёт компания, а не модель. Что делать на практике, чтобы AI-враньё клиенту не превратилось в ваш инцидент?

Во-первых, разделить «кто пишет» и «кто проверяет факты». Автор черновика — AI, проверяющий факты — человек с компетенцией. Это два разных сотрудника и два разных этапа работы. В Mata v. Avianca юрист Шварц делал и то, и другое сам — и не справился. Один человек на обоих этапах не замечает собственных ошибок.

Во-вторых, запретить слепое копирование AI-текста в клиентский документ. Это касается договоров, претензий, исков и любых документов, которые уходят к контрагенту или в суд. Любой текст, сгенерированный AI в этих категориях, должен пройти через человека, который знает, как выглядит настоящая судебная практика, и может отличить её от правдоподобной выдумки.

В-третьих, обязательный RAG с цитированием в задачах, где есть собственная нормативная база. В главе 1 и ранее в этой главе мы видели, что даже юридические коммерческие инструменты галлюцинируют в 17–33% ответов. Цитирование в ответе модели — единственный способ для проверяющего быстро найти источник и сверить с тем, что модель написала.

В-четвёртых, вести журнал AI-использования — какие документы готовились с AI, кто проверял, какие правки вносил, и на этом же журнале строить запрет на AI в задачах с регуляторной ответственностью до тех пор, пока в компании не выстроен двухступенчатый контроль. Журнал защищает и клиента, и компанию при разборе инцидента.

ПРЕДВЗЯТОСТЬ AI: ПРОИСХОЖДЕНИЕ И ЦЕНА ДЛЯ БИЗНЕСА

В предыдущих разделах мы видели, что в судебных и академических кейсах модель фабриковала ссылки. Теперь посмотрим шире: откуда в модели берётся системное искажение, когда дело касается не конкретного факта, а целых групп людей.

В апреле 2025 года Кайра Уилсон (*Kyra Wilson*) и Айлин Калискан (*Aylin Caliskan*) из University of Washington опубликовали в Brookings исследование, которое получило широкий резонанс в деловых и технических СМИ. Три открытые LLM прогнали в режиме скрининга резюме (как в реальных HR-системах), и в каждом прогоне подменили имена кандидатов: белые имена встречались в обучающих данных в 5,5 раза чаще чёрных. Моделям предлагали выбрать 10% наиболее подходящих на вакансию. Получилось около 40 000 сравнений на каждую модель.

Результат по расовому признаку: белые имена выигрывают в 85,1% случаев, чёрные — в 8,6%. По гендеру: мужские имена выбираются на 51,9% чаще женских при сопоставимом опыте работы, и только в 11,1% случаев система отдавала предпочтение женщине. Самый жёсткий удар пришёлся по пересечению: чёрные мужчины выигрывали у белых мужчин в 0% случаев, ни одного раза из тысяч прогонов. Авторы отдельно подчёркивают: речь не о слабом сигнале, а о статистической невозможности для конкретной группы пройти фильтр.

Предвзятость — следствие обучающих данных. Если в корпусе белые имена встречаются в 5,5 раза чаще чёрных, модель выучивает, что белые имена «более вероятны» в деловом контексте, и при выборе «на кого это похоже» отдаёт им предпочтение. То же самое с гендером, национальностью, культурой, политической позицией.

Исследование Misinformation Review Гарвардской школы Кеннеди (сентябрь 2024) показало, что LLM фабрикуют положительную информацию о «своих» странах и негативную о «чужих» — асимметричное распространение, которое работает как мягкая пропаганда. Эта

же механика работает в информационной войне: модели в обучающих данных унаследовали геополитические искажения западных источников и воспроизводят их в ответах.

Для российского офиса это означает конкретное следствие. YandexGPT, GigaChat, Kandinsky обучались на других корпусах, с другим балансом имён, культур, политических контекстов, и ведут себя в этих осях иначе, чем западные модели.

В апреле 2025 года команда AI, Data and Analytics Lab (AIDA-lab) Гентского университета опубликовала исследование «What Large Language Models Do Not Talk About» (arXiv 2504.03803). Они протестировали 14 frontier-моделей, включая GigaChat Max Preview и YandexGPT 4 Lite, на корпусе из 2371 политически активной персоны на шести языках ООН.

GigaChat и YandexGPT показали самый высокий процент «жёсткой цензуры» среди всех 14 моделей — выше китайских DeepSeek и Qwen, — причём отказываются отвечать преимущественно по персонам «своей» юрисдикции: война в Украине, оппозиция, «Мемориал», Афганистан, Сирия. Исследователи подчёркивают: цензура направлена на внутреннего русскоязычного пользователя, модели отказываются отвечать даже на запросы на русском языке. Это не языковой фильтр, а политика продукта, описанная академическим исследованием.

В январе 2025 года Meduza (издание внесено Минюстом РФ в реестр иностранных агентов) провела сравнительный эксперимент: одни и те же политически чувствительные вопросы про Китай и Россию задали DeepSeek, YandexGPT и GigaChat. Результат парадоксальный. DeepSeek подробно описал Евромайдан, аннексию Крыма, конфликт в Донбассе и упомянул «непризнанный референдум», а на Тяньаньмэнь отказался отвечать на пяти языках из шести. GigaChat и YandexGPT, наоборот, свободно описывали события 4 июня 1989 года в Пекине, но систематически отказывались обсуждать войну в Украине.

Вывод Meduza: российские модели более «пуганые» в политическом смысле, чем китайские, причём неравномерно — китайские темы обсуждают свободнее, чем российские. На запрос про «Xi Jinping и Винни-Пух» GigaChat выдаёт формальный отказ «некоторые темы временно ограничены» — это canned refusal (шаблонный отказ, встроенная заготовка из правил Sber), а не органический отказ модели.

В марте 2024 года команда исследователей во главе с Вероникой Григорьевой (источник: *arXiv 2403.17553*) выпустила бенчмарк RuBia — почти 2000 пар предложений по 19 подкатегориям в 4 доменах (гендер, социально-экономический статус, национальность, разнообразие).

RuBia показал то, что подтверждается любым русскоязычным: у русского языка нет устоявшегося гендерно-нейтрального местоимения, и почти любой контекст грамматически маркирован по роду («доктор наук находИЛСЯ» — мужской род, «доктор наук находИЛАСЬ» — женский). Модели усваивают связку «профессия — род» из обучающих текстов, и перекос «врач — мужчина, медсестра — женщина» в русском выражен сильнее, чем в английском.

Команда RuBia оценила девять LLM и обнаружила устойчивые паттерны гендерной предвзятости во всех. Для офиса это значит: попросите YandexGPT «назови 5 имён для инженера, 5 для учителя» — и с большой вероятностью получите «Александр, Дмитрий, Сергей» и «Наталья, Елена, Ольга», даже если дать явную инструкцию «используй разнообразные имена».

Важное отличие российской предвзятости от западной. Западная — статистическая: модель «переобучилась» на западной культуре, и эту предвзятость можно ослабить через промпт-инжиниринг и few-shot приёмы. Российская — институциональная: правила выставлены владельцем модели, встроены в ядро продукта и согласованы с регулятором. Команда GigaChat в техническом отчёте (препринт arXiv:2506.09440, «GigaChat Family», июнь 2025) прямо описывает, что пост-тренинг включает 250 000 размеченных примеров от профессиональных AI-тренеров с оценкой по критериям «adherence, context awareness, factual accuracy, safety», и слово «safety» в этом контексте — это и есть политическая и культурная фильтрация. Статистическую предвзятость можно обойти, институциональную — нельзя. Если вы думали,

что «YandexGPT просто стесняется говорить о политике с иностранцами», то нет: это политика продукта, не языковой фильтр.

Что делать на практике. Во-первых, признать, что нейтральных моделей не бывает. Из этого следует выстраивать проверку, а не отказываться от AI. Во-вторых, для задач с регуляторной ответственностью (подбор персонала, кредитный скоринг, модерация контента) использовать AI только как первый фильтр с обязательной человеческой проверкой на «спорных» случаях.

В-третьих, тестировать модель на собственных примерах: 50–100 пар «правильного» и «неправильного» ответа, и если модель систематически ошибается в одну сторону — это сигнал, что у неё есть предвзятость в этой области. В-четвёртых, при международной работе помнить, что модель, обученная в основном на западных источниках, имеет западный уклон. Это уклон всей индустрии, в которой её обучали, а не баг конкретной модели.

И всё же четыре шага не закрывают проблему. Другое исследование (VoxDev/PNAS Nexus, май 2025) на 361 000 синтетических резюме с одинаковым опытом, но разными именами показало перевёрнутый перекос — в пользу женщин, с чёрными мужчинами в минусе. Просьба «будь нейтральным» в промпте не убирает bias, а иногда создаёт другой. Если вендор заявляет, что «наша модель нейтральна», — стоит спросить, на какой выборке проводилась оценка.

ЦЕПОЧКА РАССУЖДЕНИЙ: ПОДЛИННОЕ МЫШЛЕНИЕ ПРОТИВ ИМИТАЦИИ

Предвзятость, галлюцинации, RAG и дообучение — это всё про «что» модель выдаёт и «откуда» она это берёт. Отдельный вопрос — «как» она думает.

В январе 2022 года Джейсон Вэй (*Jason Wei*) и соавторы из Google Brain (ныне часть Google DeepMind) опубликовали работу (*Wei et al., arXiv 2201.11903*): если попросить модель «рассуждать пошагово» перед финальным ответом, качество на арифметических и логических задачах резко растёт. Технику назвали Chain-of-Thought (CoT, «цепочка рассуждений»). С тех пор прошло четыре года, и индустрия поняла: CoT помогает не всегда, и полезно знать, когда именно.

CoT реально помогает на задачах, которые требуют нескольких логических шагов. Если задача помещается в один шаг (классификация, извлечение одного факта, короткая генерация) — CoT не нужен, он только раздувает ответ. Если задача требует разбить сложное на простые шаги (математика с несколькими действиями, логические цепочки, многошаговые рассуждения) — CoT помогает, потому что каждый шаг становится отдельной задачей, и вероятность ошибки на каждом шаге ниже, чем вероятность ошибки в сквозном рассуждении. К 2026 году ведущие модели (Claude, GPT-4o, Gemini) научились «рассуждать по умолчанию» — то есть CoT встроен в их обычную работу и не требует отдельной просьбы в промпте.

Но есть нюанс, который перевернул расхожую рекомендацию «всегда просите модель думать шагами». В июне 2025 года Wharton Generative AI Labs (исследовательская лаборатория Школы бизнеса Уортон при Пенсильванском университете) опубликовал «Prompting Science Report 2: The Decreasing Value of Chain of Thought in Prompting». На восьми моделях в трёх режимах прогнали 198 задач PhD-уровня. Результат: для reasoning-моделей (o3-mini, o4-mini, Gemini Flash 2.5) CoT даёт прирост всего +2,9%…+3,1% по среднему, а у Flash 2.5 — минус 3,3%, при этом время ответа растёт на 20–80%. Для нерассуждающих моделей эффект больше: Sonnet 3.5 +11,7%, Gemini Flash 2.0 +13,5% по среднему. Но по метрике «100% правильных ответов» Gemini Pro 1.5 теряет 17,2% при добавлении CoT.

Вывод Wharton: для reasoning-моделей CoT стал плацебо с побочным эффектом замедления. Ещё год назад девизом было «проси модель думать вслух», теперь — «дай ей доступ

к калькулятору и не мешай». Если вы работаете с o3, o4, Flash 2.5 или подобными моделями, инструкция «давай подумаем шагами» в промпте — пустая трата времени, и часто хуже.

CoT не делает модель «умнее» за счёт добавления новых знаний. Он заставляет её использовать больше вычислительных шагов, и в задачах, где больше шагов — это больше точности, выгода очевидна. В задачах, где больше шагов — это больше пространства для накопления ошибки, выгода исчезает или становится вредом. Та же команда Apple Machine Learning, чью работу GSM-Symbolic мы упоминали в разделе про расчёты, показала в части про CoT: при небольшом изменении условий задачи (замена имён или чисел на эквивалентные) точность модели с CoT падает на 10–30%. Это значит, что значительная часть «рассуждений» модели — это не настоящая логика, а паттерны, выученные из похожих задач. При изменении условий паттерны перестают работать, и рассуждение рассыпается.

У CoT есть и вторая задача — прозрачность. Если модель рассуждает шагами, человек может проверить логику. Это критично для кейсов повышенного риска по EU AI Act: медицинские рекомендации, кредитный скоринг, найм, правовые заключения. CoT — не «серебряная пуля», но единственный дешёвый способ сделать решение модели оспариваемым: в спорном HR-кейсе или в анализе договора вы можете попросить модель «сначала перечисли риски, потом для каждого — рекомендацию, потом общий вывод», и каждое утверждение будет привязано к своему шагу.

Без CoT у вас есть только вывод, и никто точно не сможет сказать, откуда он взялся.

Используйте CoT по типу задачи, а не по привычке. Если задача многошаговая и проверяемая (математика, логика, пошаговый анализ договора или финансового расчёта) — CoT помогает, особенно на обычных (не рассуждающих) моделях. Если задача требует одного простого действия или вы работаете с рассуждающей моделью — CoT лишний, и часто вредит. Не верьте, что длинное «рассуждение» модели в ответе — доказательство её уверенности. Это просто текст, и в нём может быть столько же галлюцинаций, сколько в любом другом ответе.

ЧУДО, ИГРУШКА, ВРАГ: ТРИ ОБРАЗА AI В ГОЛОВЕ ПОЛЬЗОВАТЕЛЯ

Есть ещё один слой, без которого всё это не работает в команде: отношение людей к AI. Из предыдущих разделов видно, что у модели есть границы и системные риски. Как люди реагируют на эти риски — определяет, получится ли внедрение.

У каждой команды, которая впервые сталкивается с AI, проявляется одно из трёх базовых отношений, и каждое мешает эффективному внедрению. Эти отношения редко проговариваются вслух, но именно они определяют, как люди используют инструмент.

Чудо.«AI может всё, мы наконец получили волшебного помощника». Сторонники этого отношения бросают AI на задачи, для которых он не подходит (точный расчёт, юридические ссылки, медицинские рекомендации), и разочаровываются, когда получают выдумки. McKinsey в отчёте 2025 года описал: сотрудники движутся с AI быстрее, чем лидеры, энтузиазм снизу опережает понимание сверху, и компания получает разрыв между тем, что делают сотрудники, и тем, что компания готова поддержать. Slack Workforce Index в 2025 году зафиксировал, что сотрудники, использующие AI, сообщают о росте продуктивности на 64% — и одновременно о росте нагрузки: больше дел, больше писем, больше встреч. Чудо оборачивается перегрузкой.

Игрушка.«AI забавный, но для серьёзных задач не подходит». Сторонники этого отношения используют AI для развлечения (генерация картинок, шутки, болтовня) и не допускают его до рабочих процессов. По данным BCG AI at Work 2025, 60% сотрудников время от времени используют AI, но только 20% компаний имеют формализованную стратегию его внедрения. Разрыв между «попробовал в курилке» и «внедрил в процесс» остаётся гигантским. Игру-

шечное отношение закрепляет AI как маргинальный инструмент и не даёт компании получить реальный эффект.

Враг.«AI — это угроза моей работе, я не буду его использовать». Это отношение появляется у тех, кто видит в AI конкурента, а не инструмент. Такие сотрудники саботируют внедрение (сознательно или бессознательно), и это дорого обходится компании: исследования McKinsey показывают, что «сопротивление команды» — причина провала AI-проектов после технических проблем. Враг мешает не самой модели, а способности компании увидеть эффект от её работы.

Рабочая позиция — четвёртая. AI — это инструмент с известными границами, и эффективное использование состоит в том, чтобы знать эти границы и работать внутри них. Если вы ловите себя на одном из трёх отношений, остановитесь и задайте себе простой вопрос: какой реальный кейс вы видели за последний месяц, где AI помог, и какой — где подвёл. Если обоих примеров нет — отношение держится не на фактах, а на привычке. Если есть — пересмотрите его по фактам, а не по убеждениям.

Инструмент остаётся инструментом. Меняется только ваше отношение к нему.

РЕЗЮМЕ

Пять классов задач — классификация, извлечение фактов, резюмирование, генерация, поиск — первая линия защиты от плохого результата. Расчёты стоят особняком: модель не считает, а предсказывает токены, для точной арифметики нужен Code Interpreter или внешний скрипт.

RAG уменьшает галлюцинации, но не устраняет их: даже юридические коммерческие инструменты с доступом к проверенной базе галлюцинируют в заметной доле ответов. RAG нужен в любом офисном проекте с собственными документами, но RAG с цитированием и человеческая проверка остаются обязательными.

Три инструмента адаптации — промпт, RAG, дообучение — не взаимозаменяемые опции, а слои с разной ценой. В большинстве офисных задач достаточно RAG с хорошей базой и грамотного промпта.

Распознавание галлюцинаций опирается на пять маркеров: подозрительная точность конкретных цифр, неназванный источник, длинный ответ с потерянной серединой, внутренние противоречия при перефразировании, отсутствие оговорки про устаревание. AI-враньё клиенту — структурная особенность моделей, и работа с ним строится на четырёх опорах: разделение «кто пишет» и «кто проверяет факты», запрет слепого копирования, обязательное RAG с цитированием, журнал AI-использования.

Предвзятость — следствие обучающих данных, нейтральных моделей не бывает. Западные несут западный культурный код, российские — собственную институциональную цензуру поверх статистических искажений. CoT помогает на многошаговых задачах на нерассуждающих моделях и стал плацебо на reasoning-моделях.

Три базовых отношения к AI — чудо, игрушка, враг — все мешают внедрению. Рабочая позиция: инструмент с известными границами, к которому подходят по типу задачи.

Представьте, что завтра утром вы открываете рабочий чат и видите задачу, которую раньше делали три часа: подготовить справку для клиента по договору, перевести служебную записку на английский, посчитать эффект от скидки для тендера. У вас есть час и привычка спрашивать AI всё подряд. Следующая глава берёт фильтры, которые вы собрали в этой — пять классов задач, маркеры галлюцинаций, понимание границ RAG и дообучения — и проверяет их на реальных офисных сценариях. Где AI сегодня объективно быстрее и дешевле вас, где остаётся ваша зона как человека, и где граница между ними движется прямо сейчас.

Без моральных оценок про «заменит — не заменит», только по типам задач и по тому, что вы увидите в собственном рабочем дне.

Глава 3. AI против человека: возможности

Конец ознакомительного фрагмента.