ПРАКТИКА ИИ · 01

ПОНЯТЬ ИИ

Понятия, механизмы и логика искусственного интеллекта простыми словами

Марина Лямкина

Марина Лямкина. «Понять ИИ».

Серия «Практика ИИ». Книга 01.

Электронное издание.

© Марина Лямкина, 2026.

Все права защищены. Не допускается воспроизведение и распространение текста и иллюстраций целиком или частично без разрешения правообладателя, кроме случаев, предусмотренных законом.

О СЕРИИ «ПРАКТИКА ИИ»

Искусственный интеллект очень быстро перестал быть отдельной темой для тех, кто специально им интересуется. Он оказался в поиске, офисных программах, телефонах, работе с документами и данными. Он пишет, считает, ищет, анализирует, создаёт изображения, помогает программировать, подключается к другим системам и постепенно забирает на себя всё более длинные цепочки работы. Вместе с возможностями растёт и количество новых слов. Модели. Контекст. Токены. Attention. RAG. API. Агенты. MCP. Fine-tuning. А рядом возникают уже совсем практические вопросы.

Что из этого действительно нужно понимать обычному человеку?

Как отличить красивый ответ от надёжного?

Что можно поручить ИИ, а что лучше оставить себе?

Когда достаточно обычного чата, а когда уже нужен отдельный помощник?

Что вообще значит «внедрить ИИ» в работу компании?

И где во всей этой конструкции остаётся человек?

Так появилась серия «Практика ИИ».

Не как энциклопедия искусственного интеллекта и не как учебный курс, после которого нужно сдать экзамен. Каждая книга отвечает на отдельный вопрос. Их можно читать по порядку, а можно брать ту, которая нужна именно сейчас.

01. Понять ИИ Что именно находится перед нами. Модели, приложения, контекст, данные, память, вероятность, ошибки, поиск, инструменты — и как всё это связано.
02. Работать с ИИ Как перейти от случайных запросов к нормальной работе. Как ставить задачи, давать данные, анализировать материалы, проверять результат, вести проекты и постепенно собирать собственного рабочего помощника.

03. Управлять с ИИ Что происходит с управлением, когда часть интеллектуальной работы можно делегировать машине. Решения, постановка задач, контроль, границы делегирования и ответственность.
04. Внедрять ИИ Как отдельная удачная работа с нейросетью превращается в устойчивый процесс.тЧто стоит автоматизировать, где нужен человек и почему внедрение начинается не с выбора технологии.
05. Бизнес с ИИ Где искусственный интеллект действительно создаёт ценность. Как отличать экономический эффект от красивой демонстрации технологии и понимать, за что бизнесу вообще имеет смысл платить.
06. Создавать с ИИ Как использовать ИИ не только внутри существующей работы, но и для создания нового. От идеи и прототипа до собственного цифрового решения.

У серии есть один общий принцип: сложные понятия и процессы можно объяснять простыми словами без того, чтобы превращать их в примитивные. Именно с этого и начинается первая книга.

ОТ АВТОРА

Почему я решила начать с понимания

Я довольно давно и много работаю с искусственным интеллектом. Использую его сама, обучаю людей, разбираю вместе с компаниями рабочие задачи, читаю документацию, постоянно пробую что-то новое. И при этом у меня до сих пор регулярно бывает ситуация: встречаю очередное название или термин и думаю:

— Так. А это теперь что?

Потом начинаю разбираться. Одно определение ведёт к другому. В нём ещё три незнакомых слова. Открываешь следующую вкладку, потом ещё одну — и через какое-то время уже плохо помнишь, с какого вопроса всё началось. Причём это происходит не только с какими-то особенно техническими вещами. Казалось бы, обычные слова. Модель.

Память. Контекст. Агент. Поиск. Всё звучит знакомо, пока не пытаешься точно понять, что именно человек имеет в виду.

Если такое регулярно происходит со мной, хотя ИИ — часть моей работы, я хорошо представляю, каково человеку, который занимается совсем другим делом и просто пытается разобраться, что происходит вокруг. Ещё несколько лет назад в это можно было особенно не погружаться. Неинтересно — и ладно. Сейчас так становится всё сложнее. Искусственный интеллект постепенно появляется в программах, которыми мы пользуемся каждый день, в поиске, документах, аналитике, продажах, обучении, маркетинге, разработке.

Всё чаще вопрос уже не в том, интересует вас ИИ или нет. Он просто оказывается рядом. При этом я совершенно не считаю, что теперь всем срочно нужно становиться специалистами по машинному обучению. Мне, например, не кажется необходимым знать устройство двигателя, чтобы нормально водить машину. Но всё-таки очень полезно понимать, что именно находится перед тобой.

Почему ChatGPT — это не то же самое, что модель GPT. Почему один и тот же сервис предлагает несколько моделей. Почему модель может прекрасно объяснить сложную тему и через минуту придумать несуществующий факт. Что означает контекст и почему его иногда не хватает. Почему ИИ вдруг «забывает» то, что вы обсуждали вчера.

Что значит большое контекстное окно. Почему в одном случае ИИ действительно сам формирует ответ, а в другом сначала ищет информацию в интернете, читает файл или вызывает калькулятор.

И почему фраза:

— Мы хотим внедрить ИИ,

сама по себе почти ничего не говорит о том, что именно компания собирается делать. Первая книга серии появилась именно отсюда.

Не потому, что это «самый простой уровень».

И не потому, что сначала нужно выучить двадцать терминов, получить право перейти к следующей книге и только потом начать работать. Наоборот. Большинство людей начинает работать с ИИ задолго до того, как узнаёт, что такое LLM или токен. И это нормально. Проблема возникает позже.

Когда результат оказывается странным. Когда один сервис делает то, чего почему-то не делает другой. Когда ИИ уверенно сообщает неверную информацию.

Когда человек загружает огромный файл и уверен, что теперь модель «его знает».

Когда в компании обсуждают RAG, агента или API и часть людей кивает скорее из вежливости. Тогда понимание начинает экономить очень много времени. Я не хочу прятать от читателя профессиональные термины. Если слово действительно используется в документации, на конференциях, в интерфейсах или в разговоре разработчиков, лучше знать, как оно

называется. Но мне ещё меньше хочется объяснять один непонятный термин пятью другими.

Поэтому здесь будет два уровня. Сначала — что это вообще значит. Потом, если понятие действительно важно, — как оно называется профессионально и что происходит чуть глубже.

Например, мне недостаточно написать:

LLM — это Large Language Model, большая языковая модель. Формально всё правильно. Только человеку, который впервые встретил аббревиатуру LLM, от расшифровки стало ненамного легче.

Гораздо интереснее понять:

что именно в этой конструкции является моделью;

почему она языковая;

почему она большая;

как вообще получается, что одной и той же модели можно сначала дать договор, потом попросить найти условия оплаты, затем написать письмо клиенту, а после этого попросить объяснить сложный пункт договора простыми словами. То же самое с attention. Можно вообще никогда не вспоминать Query, Key и Value и прекрасно пользоваться ИИ. Но полезно хотя бы один раз понять, почему слова и фрагменты длинного текста для модели не существуют независимо друг от друга и каким образом контекст влияет на то, какой ответ появится дальше. Или взять промпт.

Само слово успело обрасти таким количеством инструкций, шаблонов и «секретных техник», что временами создаётся впечатление, будто человеку сначала нужно научиться разговаривать на специальном языке. На практике очень часто всё начинается гораздо прозаичнее. Нужно самому понимать, что вы хотите сделать. Что дать модели. Какой результат получить.

И что в этом результате нельзя додумывать. Вторая книга серии будет уже именно об этом — о работе. Но сначала я хочу собрать фундамент.

Чтобы слова перестали мешать видеть сам процесс. При этом после этой книги у вас всё равно останутся вопросы.

У меня они тоже остаются. Более того, в этой области регулярно появляются новые. И мне кажется, это совершенно нормальное состояние. Здесь слишком быстро всё меняется, чтобы однажды «разобраться в ИИ» и поставить галочку. Но есть большая разница между двумя состояниями.

Первое:

Я вообще не понимаю, о чём они сейчас говорят.

И второе:

Вот это я понимаю. А вот здесь мне нужно уточнить. Мне хочется, чтобы первая книга серии помогла перейти из первого во второе. Не знать всё. Не запоминать каждое сокращение. Не учиться изображать специалиста по машинному обучению.

А достаточно хорошо понимать устройство происходящего, чтобы не работать с ИИ вслепую. И когда через месяц появится очередной термин, не пугаться того, что вы его ещё не знаете. А просто разобраться.

КАК ЧИТАТЬ ЭТУ КНИГУ

Эту книгу можно читать подряд. А можно возвращаться к ней как к справочнику, когда в очередной статье, презентации или разговоре появляется слово, смысл которого вроде бы понятен — но не совсем. Если тема ИИ для вас пока новая, я бы всё-таки начала с начала. Понятия здесь специально стоят в определённом порядке. Сначала разберёмся, что вообще называют искусственным интеллектом, что такое модель и чем она отличается от приложения.

Потом появятся LLM, токены, attention. После этого будет гораздо проще говорить о промпте, контексте, памяти и длинном диалоге. Дальше разберёмся, почему ИИ может ошибаться, что такое галлюцинации, как проверять факты и почему наличие красивой ссылки ещё не гарантирует наличие источника. Затем посмотрим, что меняется, когда у модели появляются файлы, интернет-поиск и внешние инструменты. А ближе к концу соберём всё это в короткую карту: что умеют современные ИИ-системы и из каких частей складываются эти возможности.

Если вы уже давно пользуетесь ИИ, необязательно читать книгу от первой страницы до последней. Возможно, вам интереснее сразу открыть контекст, память, галлюцинации, поиск или инструменты. Если внутри раздела встретится понятие, которое мы уже разбирали раньше, всегда можно вернуться назад. И ещё одна просьба. Не пытайтесь запомнить все термины.

Это не тот результат, ради которого написана книга. Если через неделю вы забудете точное определение токена, ничего страшного. Гораздо полезнее помнить, что токен — не обязательно слово и что именно в токенах измеряется контекстное окно. Если не сможете воспроизвести формулу self-attention, тоже ничего страшного. Важно понимать сам принцип: модель учитывает связи между разными частями контекста, а не читает каждое слово как независимый элемент.

Если забудете полную расшифровку какого-нибудь сокращения, её всегда можно посмотреть. Мне важнее другое.

Чтобы, столкнувшись с новым термином, вы могли задать несколько нормальных вопросов:

— Что именно это делает?

— Что поступает на вход?

— Что получается на выходе?

— Это делает сама модель или другая часть системы?

— Откуда берутся данные?

— Что здесь может пойти не так?

— Где нужен человек?

После таких вопросов большая часть магии довольно быстро исчезает.

Подробную работу с текстами, данными, исследованиями, проектами, помощниками, RAG, автоматизацией и агентами я вынесла во вторую книгу серии — «Работать с ИИ».

Не потому, что всё это «слишком сложно» для первой. Просто это другой вопрос.

Здесь мы разбираемся:

что именно происходит?

Там будем разбираться:

как этим нормально пользоваться в работе?

А пока начнём с самого первого слова. И, как выяснится уже через несколько страниц, даже с ним всё не так очевидно. Искусственный интеллект.

ЧАСТЬ I

Давайте сначала разберёмся, кто есть кто

Когда я начинаю обучение по искусственному интеллекту, довольно быстро выясняется одна интересная вещь. Люди уже пользуются ИИ, но одни и те же слова понимают совершенно по-разному.

Кто-то говорит:

— Я работаю в ChatGPT.

Кто-то:

— Я использую GPT.

Кто-то:

— Я всё делаю в нейросети.

Ещё можно услышать:

— Какая у тебя нейронка?

— А эта модель умеет интернет?

— Я обучила себе ChatGPT.

— Алиса — это тоже нейросеть?

И чаще всего из контекста вполне понятно, что человек хочет сказать. В обычном разговоре нет ничего страшного в том, чтобы назвать ChatGPT нейросетью. Проблемы начинаются тогда, когда нужно понять, почему один инструмент умеет то, чего не умеет другой, что именно мы выбираем в списке моделей и почему после переключения какого-нибудь пункта в интерфейсе результат вдруг становится другим. Тут уже оказывается полезно хотя бы немного развести понятия. Не все. Я точно не собираюсь в самом начале заставлять вас разбираться в архитектурах нейронных сетей, типах обучения и количестве параметров. До большей части этих

вещей обычный пользователь может вообще никогда не дойти и прекрасно себя чувствовать.

Но несколько слов нам всё-таки понадобятся. В первую очередь — искусственный интеллект, модель, языковая модель и приложение. Потому что сегодня они постоянно смешиваются. Например, ChatGPT часто называют моделью. Хотя ChatGPT — это продукт, приложение, которым мы пользуемся. А внутри него работает модель — и не обязательно всегда одна и та же.

С GigaChat ситуация ещё веселее: одним словом могут называть и пользовательский сервис, и семейство моделей. У Яндекса есть Алиса как продукт, а есть модели, которые работают внутри экосистемы Яндекса. У Google словом Gemini называется и пользовательский продукт, и семейство моделей. То есть запутаться здесь вполне реально. И дело не в том, что вы «недостаточно технический человек». Сами названия устроены так, что временами будто специально помогают всё смешать. Я бы сначала собрала самую простую карту.

Не для того, чтобы потом разговаривать исключительно правильными техническими терминами. А для того, чтобы понимать, что именно находится перед нами.

Очень грубо наша карта будет выглядеть так:

Искусственный интеллект — большая область. Внутри неё существуют разные подходы и модели. LLM — один из классов современных моделей, работающих с языком. Есть конкретные модели конкретных разработчиков. А уже вокруг моделей создаются приложения и сервисы, которыми пользуемся мы.

И правда: для первого захода этого достаточно. К нейросетям, генеративному ИИ и другим словам будем возвращаться только тогда, когда они действительно понадобятся. А начнём всё-таки с самого первого. С искусственного интеллекта. Потому что оказалось, что слово,

которым мы сегодня называем всё подряд, само по себе не такое уж простое.

Искусственный интеллект

Начнём, пожалуй, с самого неприятного вопроса. Что такое искусственный интеллект? Не «как им пользоваться», не «какую нейросеть выбрать», а вообще — что мы называем искусственным интеллектом. Потому что сейчас этим словом называют примерно всё. ChatGPT — ИИ. Система, которая рекомендует вам фильм, — тоже ИИ. Камера на производстве, которая замечает брак, — ИИ. Банк блокирует подозрительную операцию — опять где-то рядом ИИ. А потом появляется кофеварка с надписью AI, и становится совсем сложно понять, где проходит граница.

ОПРЕДЕЛЕНИЕ

> Искусственный интеллект — область методов и технологий, с помощью которых создают машинные системы, способные на основании входных данных выводить, какой результат нужно сформировать: прогноз, классификацию, рекомендацию, решение, текст, изображение и так далее. Источник: [1]

Определение точное. Дружелюбным я бы его не назвала.

Но я бы его не выбрасывала и не заменяла фразой вроде:

«ИИ — это компьютер, который умеет думать как человек».

Потому что вот это как раз неправда.

Более того, одна из вещей, которые очень полезно понять в самом начале:

ИИ не думает так, как думаем мы. Хотя делает всё возможное, чтобы нам так казалось.

Вы задаёте вопрос:

— Как ты думаешь, почему клиент отказался?

И получаете:

— Я думаю, здесь может быть несколько причин.

Очень человеческий разговор. Мозг моментально достраивает картину: мой вопрос прочитали, поняли, обдумали, сформировали мнение и теперь мне его рассказывают. Но никакого маленького цифрового сотрудника, который сидит внутри компьютера, там нет. В основе того генеративного ИИ, с которым большинство из нас сегодня имеет дело, находится математическая модель. Огромная, очень сложная, с большим количеством параметров, настроенных в процессе обучения на данных. Слово «математическая» иногда пугает сильнее самого «искусственного интеллекта». И зря.

Формулы здесь не понадобятся. Для нашей задачи сейчас достаточно понять сам принцип. Модель обучалась на большом количестве данных и в процессе обучения находила в них закономерности. В случае языковой модели — закономерности языка: какие элементы текста связаны друг с другом, какие продолжения возникают в определённом контексте, какие конструкции обычно следуют за другими. После обучения у нас появляется модель с настроенными числовыми параметрами.

А потом приходим мы.

И пишем:

Например: «Напиши письмо клиенту». Модель что-то ответит.

Теперь добавим:

Мы задержали поставку на три дня. Клиент уже раздражён. Нужно короткое письмо: признать задержку, назвать новый срок, не перекладывать ответственность на логистов и предложить созвон, если новый срок для него критичен. Модель та же самая. Но результат уже другой. Появился контекст.

Вот это я бы поняла раньше любого «промпт-инжиниринга»:

Результат работы ИИ зависит не только от самой модели. Он зависит ещё и от того, что именно мы ей дали.

Если очень грубо собрать происходящее в одну схему, получится примерно так:

МОДЕЛЬ + ТО, ЧЕМУ ОНА НАУЧИЛАСЬ НА ДАННЫХ + ТЕКУЩИЙ КОНТЕКСТ

↓

ВЫЧИСЛЕНИЯ

↓

РЕЗУЛЬТАТ

Но в этой схеме пока не хватает одного очень важного слова. Вероятность. К вероятности мы ещё вернёмся отдельно. Пока достаточно заметить саму странность: модель способна написать очень убедительный текст, а убедительность ещё ничего не говорит о том, проверен ли факт. А можно тогда вообще говорить, что ИИ «думает»? Можно.

Я сама дальше в книге буду писать:

«ИИ думает».

«ИИ понимает».

«ИИ решил».

«ИИ предлагает».

Не потому, что мы сейчас договорились считать математическую модель человеком.

Просто фраза:

«Модель на основании текущего контекста и распределения вероятностей сформировала последовательность токенов…»

быстро убьёт любую книгу. Поэтому человеческий язык оставим. Но будем помнить, что это удобная метафора. Когда мы говорим «ИИ понял задачу», это не означает, что внутри произошло человеческое понимание.

Когда говорим «ИИ решил», это не означает, что он сел, подумал, взвесил последствия и принял решение так, как это сделал бы человек.

Это всего лишь нормальный человеческий язык для очень сложного технического процесса. И я собираюсь им пользоваться.

Как это выглядит в жизни

Здесь как раз удобно вернуться к тем продуктам, которыми мы реально пользуемся. Когда вы открываете ChatGPT, GigaChat, Алису или Gemini, вы видите не математическую модель. Вы видите готовое приложение. Внутри него может работать одна или несколько моделей. А вокруг модели могут быть подключены другие возможности: поиск в интернете, работа с файлами, память, генерация изображений, вычисления и другие инструменты.

Отсюда и знакомая фраза:

«ИИ нашёл информацию в интернете».

Но технически происходить могло другое:

`приложение запустило поиск → нашло страницы → передало их содержимое модели → модель сформировала ответ.`

Или:

«ИИ посчитал».

А на самом деле модель вызвала отдельный вычислительный инструмент.

Или:

«ИИ помнит, что я ему рассказывала».

А память реализована как отдельная функция приложения.

Поэтому дальше мы будем постепенно разделять:

что умеет сама модель;

и что умеет приложение вокруг неё. Если эту границу поймать в самом начале, дальше становится заметно проще.

Нейросеть и генеративный ИИ

До этого я несколько раз употребила слово «нейросеть» почти так же, как его обычно употребляют в разговоре: «открой нейросеть», «спроси у нейросети», «какая нейронка лучше». В быту это понятно. Но если мы уже начали раскладывать конструкцию по частям, здесь стоит сделать ещё одно короткое уточнение.

ОПРЕДЕЛЕНИЕ

> Нейронная сеть — один из типов вычислительных моделей машинного обучения. Она состоит из большого количества связанных между собой вычислительных элементов и настраивается на данных. Современные большие языковые модели построены на нейросетевых архитектурах. Источник: [2]. Но слово «нейросеть» не равно слову «искусственный интеллект». И тем более оно не означает конкретное приложение. ChatGPT можно в бытовом разговоре назвать нейросетью, но технически перед нами приложение, внутри которого работают модели, а вокруг них — дополнительные функции.

ОПРЕДЕЛЕНИЕ

> Генеративным называют ИИ, который создаёт новый результат на основе усвоенных закономерностей: текст, изображение, аудио, код и другие виды содержимого. Это не весь искусственный интеллект. Система, которая только классифицирует брак на производстве или рассчитывает риск по заданной модели, тоже может относиться к ИИ, но её задача не обязательно генеративная. Источник: [5]. LLM — один из классов генеративных моделей: они работают с языком и формируют новые последовательности текста. Поэтому в этой книге слова «ИИ», «нейросеть», «генеративный ИИ», «LLM» и «приложение» не будут

использоваться как полные синонимы. В обычном разговоре это не преступление. Для понимания механики — уже полезное различие.

Теперь карта становится чуть точнее: искусственный интеллект — большая область; нейронная сеть — тип модели; генеративный ИИ описывает системы, которые создают новый результат; LLM — языковые генеративные модели; а приложение — то, с чем непосредственно работает пользователь.

Модель

Это слово будет встречаться в книге постоянно, поэтому с ним лучше разобраться сразу.

В обычном разговоре мы легко смешиваем:

«я работаю в ChatGPT»
«я выбрала другую модель»
«эта нейросеть отвечает лучше»
«у меня включён GPT»

И примерно понимаем друг друга. Но технически это всё-таки разные вещи. Мне нравится простая аналогия: модель — двигатель, приложение — автомобиль. Она не идеальна, но для первого знакомства работает хорошо.

ОПРЕДЕЛЕНИЕ

> Модель — формализованная вычислительная система с набором параметров, настроенных в процессе обучения, которая преобразует входные данные в выходной результат. Источник: [2][3]

Модель — это та часть ИИ-системы, которая непосредственно выполняет вычисления. Вы что-то даёте ей на вход. Она обрабатывает этот вход с учётом того, чему научилась раньше. И выдаёт результат. Для языковой модели входом может быть ваш текст, предыдущая переписка, инструкции, содержимое файла.

Результатом — текст, таблица, классификация, структура данных или другой ответ. При этом модель — ещё не всё приложение, которым вы пользуетесь. И вот это различие новичку действительно важно.

Почему ChatGPT — не модель

Когда вы открываете ChatGPT, перед вами не просто GPT-модель. Перед вами целый продукт. В нём есть интерфейс. История разговоров. Работа с файлами.

Поиск. Память. Инструменты. Возможность переключать режимы и модели. То есть модель находится внутри системы, но пользователь работает со всей системой целиком.

Поэтому технически точнее говорить:

ChatGPT — приложение. GPT — семейство моделей, которые могут использоваться внутри этого приложения. То же самое происходит и с другими продуктами. Например, у Сбера есть пользовательский продукт GigaChat, а внутри экосистемы есть конкретные модели GigaChat разных поколений и классов. У Google словом Gemini называется и пользовательский продукт, и семейство моделей, что делает ситуацию ещё менее очевидной. У Яндекса пользователь может общаться с Алисой, но сама Алиса — это уже продукт, внутри которого используются модели и другие компоненты.

Поэтому фраза:

«Какая у тебя нейросеть?»

в быту вполне понятна.

Но если мы хотим разобраться, что именно сравниваем, лучше спросить:

Какой сервис ты используешь?

и отдельно:

Какая модель работает внутри?

Зачем вообще знать название модели

Может возникнуть вполне разумный вопрос:

А мне какая разница? Главное, чтобы отвечало хорошо. Во многих бытовых задачах — действительно никакой. Если вам нужно придумать поздравление, сократить письмо или составить меню на неделю, совершенно необязательно знать, какая именно модель сейчас работает. Но разница появляется, когда задача становится сложнее. Одна модель может лучше работать с большим текстом.

Другая — быстрее отвечать. Третья — лучше справляться с кодом. Одна может поддерживать изображения. Другая — длинные документы. Третья может быть дешевле для массовой обработки.

Поэтому внутри одного и того же сервиса разработчик иногда предлагает несколько моделей. На этом я бы остановилась. Названия, номера версий и внутренние линейки меняются слишком быстро, чтобы превращать первую книгу серии в каталог. Полезнее помнить принцип: приложение и модель — разные уровни, а внутри одного приложения моделей может быть несколько.

И ещё одна вещь, о которой легко забыть

Даже если две системы используют одну и ту же базовую модель, результат у пользователя может отличаться. Почему?

Потому что вокруг модели есть ещё очень многое:

инструкции разработчика;

поиск;

работа с файлами;

дополнительные инструменты;

память;

ограничения;

настройки самого приложения.

Поэтому:

одна и та же модель внутри разных систем — не всегда один и тот же пользовательский опыт.

И это ещё одна причина, почему я не люблю вопрос:

«Какая нейросеть лучше?»

Он слишком широкий.

Как это выглядит в работе

Очень грубо можно представить так:

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ
большая область

↓

КЛАСС МОДЕЛЕЙ
например языковые модели

↓

СЕМЕЙСТВО МОДЕЛЕЙ
GPT, Claude, Gemini, GigaChat и др.

↓

КОНКРЕТНАЯ МОДЕЛЬ
определённая версия внутри семейства

↓

ПРИЛОЖЕНИЕ
то, что открывает пользователь

↓

ПОИСК, ФАЙЛЫ, ПАМЯТЬ, ИНСТРУМЕНТЫ И ДРУГИЕ ФУНКЦИИ

Эту схему стоит запомнить, а конкретные модели воспринимать как справку «актуально на дату издания»: они устаревают значительно быстрее самого принципа. А теперь можно двигаться дальше.

Потому что мы несколько раз употребили выражение «языковая модель».

И здесь возникает следующий вполне логичный вопрос:

Что такое LLM — и почему именно с ней мы разговариваем почти как с человеком?

Большая языковая модель — LLM

С этим сокращением вы будете встречаться постоянно. LLM — Large Language Model. По-русски — большая языковая модель. Слово «модель» мы уже разобрали. Теперь осталось понять, что в ней языкового и почему она большая.

ОПРЕДЕЛЕНИЕ

> Большая языковая модель, или LLM, — языковая модель большого масштаба, обученная на больших массивах последовательностей текста оценивать вероятное продолжение и генерировать новые последовательности с учётом контекста. Большинство современных LLM используют архитектуру Transformer или её производные.
>
> Источники: [3][4].

Определение правильное, но если вы встретили его впервые, понятнее от него стало не намного. Поэтому разберём без спешки. LLM — это модель, которую обучили настолько хорошо работать с языком, что мы можем ставить ей очень разные задачи обычными словами.

Например, дать стенограмму встречи:

Выдели все решения. Для каждого укажи ответственного и срок. Если ответственный или срок прямо не назван, не придумывай.

Потом в том же диалоге попросить:

Теперь сделай из этого короткое письмо участникам встречи.

А потом:

Слишком официально. Напиши проще. Мы не переключали три разных программы. Не переобучали модель. Не писали код. Мы просто меняли задачу словами.

И вот это для обычного пользователя намного важнее, чем устройство Transformer.

Почему она языковая

Потому что язык — её основной рабочий материал. Тексты, вопросы, ответы, документы, переписка, инструкции, программный код — всё это для языковой модели последовательности, внутри которых она научилась находить связи и закономерности. Поэтому она может не только написать текст. Она может работать с уже существующим.

Например, у нас есть письмо клиента:

Нас в целом устраивает предложение, но запуск проекта в сентябре невозможен. До конца октября мы заняты переносом производства. Вернуться к вопросу готовы в ноябре. Перед этим хотелось бы получить уточнённый расчёт окупаемости.

Можно спросить:

Что здесь важно для менеджера по продажам?

И получить примерно такую структуру:

предложение клиенту интересно;

запуск в сентябре невозможен;

причина — перенос производства;

вернуться к обсуждению готовы в ноябре;

до следующего контакта нужен уточнённый расчёт окупаемости.

Вот здесь особенно хорошо видно, почему слово «языковая» не означает просто «умеет сочинять тексты».