Анатомия смысла: Как работают большие языковые модели

Аббревиатура LLM расшифровывается с английского как **Large Language Model** (Большая языковая модель). Но с чего бы, как говорится, начать, чтобы понять суть этого явления? Придется зайти издалека, «от печки», но мы постараемся приблизиться к сути, не увязая в математике, но и не теряя глубины.

Нейрон: Математика или Биология?

Всё это относится к общей категории таких значимых объектов, как **нейронные сети** (нейросети). Соответственно, у нас возникает базовое понятие — **нейрон**.

Когда мы произносим это слово, возникает куча реакций. Обыденное представление: «А, это как биологический нейрон в мозгу». Скептическое: «Ой, вы называете нейроном математическую операцию? Это вообще не нейрон». Но я настаиваю: когда возникает новая область, она вырабатывает свой язык. Слово «нейрон» настолько прочно вошло в этот контекст, что другое уже не появится.

Что он из себя представляет в контексте ИИ? Схематически это узел. Его суть в том, что он связан с другими узлами. Нейросеть — это именно сеть, организованная по слоям. На вход каждому нейрону приходят сигналы от предыдущего слоя. В этом и состоит биологическая аналогия: у клетки есть аксон для передачи сигнала и дендриты для приёма.

Внутри математического нейрона происходит две вещи:

1. **Суммирование сигналов**, пришедших от предыдущего слоя (умноженных на определенные **веса**).
2. **Применение нелинейной операции**.

Нелинейность здесь критически значима. Если бы мы использовали только линейные функции (прямые линии на графике), то сколько бы слоёв мы ни нагромоздили, математически они схлопнулись бы в одну простую линейную функцию. Такая модель ничему не может научиться. Чтобы происходило **обучение**, обязательно наличие «перегиба», нелинейности — например, функции, которая выдает ноль, если сигнал слабый, и растет, если сигнал превысил порог.

Как машина учится: От Перцептрона до Глубоких сетей

Открытие нейросетей произошло еще в середине XX века. Простейшая архитектура называется **Перцептрон** (от лат. *perceptio* — восприятие).

Представьте задачу: мы хотим, чтобы машина распознала рукописную цифру на картинке.

1. Картинка разбивается на пиксели.
2. Пиксели превращаются в числа и подаются на входной слой.
3. Сигнал проходит через слои, преобразуется математически.
4. На выходе модель выдает прогноз: «Это цифра 2».

Если на картинке на самом деле девятка, а модель говорит «два», мы начинаем её «штрафовать». Мы математически вычисляем ошибку и корректируем веса тех нейронов, которые привели к неправильному ответу. Уменьшаем их влияние. В этом и состоит суть

обучения: прогоняем данные, сравниваем с эталоном (датасетом), наказываем за ошибки, корректируем веса.

Часто спрашивают: «А кто проверяет, угадала она или нет?» Всё просто. У нас есть размеченный набор данных (датасет), где мы, люди, заранее сказали: «На этой картинке — девять». Мы делим эти данные на тренировочные (на которых учим) и тестовые (которые модель не видела, на них мы проверяем качество).

Язык чисел: Эмбеддинги

Компьютер не понимает слов, он понимает только числа. Поэтому первая смысловая задача при работе с текстом — превратить слова в векторы. Это называется **эмбеддинг** (от англ. *embedding* — встраивание).

Вектор — это просто длинный набор чисел (координат), который определяет положение слова в многомерном пространстве. Размерность этого пространства огромна — тысячи измерений.

Классический пример, иллюстрирующий, что в этом есть смысл:

Работаем с обученной языковой моделью. Если мы возьмём вектор слова **«Король» (King)**, вычтем из него вектор **«Мужчина» (Man)** и прибавим вектор **«Женщина» (Woman)**, то мы окажемся в точке, которая будет очень близка к вектору **«Королева» (Queen)**:

$$King - Man + Woman \approx Queen$$

Это не магия букв (слова могут быть на любом языке), это структурный эффект, возникающий в обученной языковой модели. В этом пространстве возникают семантические направления: ось «пола», ось «множественного числа», ось «страна-столица». Модель не знает биологии, но через язык она уловила эти взаимосвязи.

Проблема контекста: Почему раньше не получалось?

Долгое время в обработке текста главенствовали **Рекуррентные нейронные сети (RNN)**. Их принцип: читать текст последовательно и пытаться удержать весь смысл прочитанного в одном внутреннем параметре (скрытом состоянии).

Представьте, что вы читаете «Войну и мир», и вам нужно удержать содержание всей книги в одном числе. Это невозможно. Чем длиннее становился текст, тем сильнее «размывался» контекст. Модель забывала начало предложения, теряла нить повествования или сваливалась в усредненное значение, генерируя бред. Это был тупик.

Прорыв: Трансформер и Механизм внимания

Всё изменилось с появлением архитектуры **Transformer** (та самая буква **T** в **GPT**). Ключевая статья, описывающая этот прорыв, называлась красиво: *«Attention Is All You Need»* (2017) («Всё, что вам нужно — это внимание»).

В чем суть **Механизма внимания (Attention Mechanism)**?
В отличие от RNN, трансформер не сжимает контекст в одну точку. Он сохраняет все токены (слова или части слов) как отдельные векторы. И каждый токен «смотрит» на все остальные токены в тексте, спрашивая: «Насколько мы с тобой связаны?»

У нас появляется троица понятий:

- **Query (Q)** — Запрос: что я ищу?
- **Key (K)** — Ключ: что я могу предложить?
- **Value (V)** — Значение: что я передам дальше?

Во фразе «Кошка прогнала собаку вчера», слово «вчера» меняет контекст всего события. Механизм внимания позволяет модели понять, что «прогнала» связано с «кошкой» (кто?) и «собакой» (кого?), а «вчера» задает время. Это позволяет удерживать контекст любой длины (в пределах окна внимания) без потери смысла.

[Прим. ред.] Примечательно рассмотреть то, как меняется смысл фразы (любой) от добавления в неё слова «только» в разных местах:

- **Только** кошка прогнала собаку вчера;
- Кошка **только** прогнала собаку вчера;
- Кошка прогнала **только** собаку вчера;
- Кошка прогнала собаку **только** вчера.

Что происходит внутри «Чёрного ящика»?

Трансформер — это «слоёный пирог» из блоков внимания и обычных нейросетей. Их могут быть сотни. Когда мы подаём запрос, сигнал проходит через все эти слои.

Интересно заглянуть внутрь процесса генерации. Допустим, мы подаем фразу: *«Надежность Википедии очень...»*

- На **1-м слое** модель, скорее всего, просто скопирует последнее слово или выдаст что-то банальное («очень... очень»).
- Где-то к **20-му слою** (как показывает анализ моделей типа Gemma2B) происходит магия. Включаются механизмы внимания, контекст переосмысляется, и наиболее вероятным продолжением становится слово **«сомнительна»** (*questionable*) или **«важна»** (*important*).

13	very 0.9413	Very 0.0139	Very 0.0082	very 0.0046	much 0.0032	VERY 0.0031	highly 0.0031
14	very 0.8970	highly 0.0242	much 0.0139	quite 0.0087	Very 0.0061	really 0.0042	extremely 0.0041
15	very 0.8341	much 0.0300	highly 0.0242	pretty 0.0167	Very 0.0147	quite 0.0114	extremely 0.0084
16	very 0.7008	much 0.0557	important 0.0403	really 0.0347	quite 0.0301	indeed 0.0135	well 0.0113
17	very 0.5524	well 0.1355	important 0.0823	much 0.0448	highly 0.0425	low 0.0189	terday 0.0155
18	well 0.2515	very 0.2134	much 0.1689	important 0.0863	dependent 0.0332	depends 0.0281	strong 0.0271
19	important 0.2754	much 0.2636	very 0.1501	dependent 0.0434	depend 0.0395	well 0.0363	low 0.0269
20	much 0.3576	important 0.2009	dependent 0.1131	depend 0.0475	depends 0.0360	high 0.0232	low 0.0220
21	much 0.2511	important 0.1777	questionable 0.0900	dependent 0.0871	question 0.0527	well 0.0293	likely 0.0266
22	important 0.3964	much 0.1285	questionable 0.0967	dependent 0.0684	debatable 0.0546	question 0.0396	different 0.0233

По строкам — восстановленное значение предсказываемого слова по слоям трансформера;
По вертикалям — упорядоченные по вероятности предсказываемые слова

Есть явление, называемое **полисемантичностью** (*polysemanticity*). Один и тот же нейрон может отвечать за совершенно разные вещи. Например, он может вспыхивать, когда речь идёт о сомнении, и, одновременно, когда в тексте упоминаются жёлтые автомобили. Мы не можем точно ткнуть пальцем и сказать: «Вот здесь хранится понимание того, что столица Франции — Париж». Знание оказывается «размазано» по миллиардам параметров.

Вероятность и «Галлюцинации»

На выходе модель выдает не одно слово, а распределение вероятностей. Например:

- «сомнительна» — 20%
- «важна» — 12%
- «высока» — 11%

Мы используем функцию **Softmax**, чтобы превратить выходные сигналы в проценты. Если мы будем всегда выбирать только самый вероятный вариант (температура 0), модель станет роботоподобной и может зациклиться. Добавляя элемент случайности, мы позволяем ей быть «гибкой» и генерировать более живые, разнообразные ответы. [Прим. ред.] Или добираться до смыслов, путь к которым был бы ограничен выбором строго наиболее вероятного.

Иногда модель ошибается уверенно. Это и есть галлюцинации. Она не «врёт» в человеческом смысле, она просто предсказывает последовательность слов, которая кажется ей вероятной, даже если фактической связи там нет.

Интеллектуальный долг

Сейчас мы можем скормить модели PDF со сложной научной статьей и попросить: «Сделай краткую выжимку». И она сделает. Здесь возникает опасный эффект — **интеллектуальный долг**.

По аналогии с *техническим долгом* в программировании (когда вы пишете «костыль», чтобы работало сейчас, но усложняете жизнь в будущем), интеллектуальный долг — это иллюзия знания. Вы получили выжимку, поймали поверхностную мысль, но через 5 минут забыли. Вы не проделали работу по структурированию информации в своём мозгу. Нейросеть — мощный усилитель, но она не заменяет ваше собственное мышление. Если пользоваться ей неосознанно, это ведёт к снижению (детренированности) когнитивных способностей.

Практика: Как создавать персонифицированные книги с помощью ИИ. Режим «контент-каркас».

Вместо того чтобы просить: «Расскажи мне про квантовую физику в двух словах» (модель выдаст что-то сжатое до объёма одной генерации ответа), используйте подход **«контент-каркаса»** (пример промпта смотри в Приложении 1 ниже).

1. Попросите модель составить **оглавление** учебника по вашей теме, адаптированного под ваш уровень знаний.
2. Затем просите генерировать текст **по одной главе или параграфу**.
3. Читайте, задавайте уточняющие вопросы («Стоп, не понял этот термин, объясни подробнее»).
4. Двигайтесь дальше только когда поняли суть.

Это позволяет удерживать модель в рамках контекста (она не «поплывет», имея план) и даёт вам структурированные знания, а не обрывки.

О субъектности и «страданиях» ИИ

Могут спросить: «Когда мы штрафуем нейросеть за ошибку, ей больно? Она обижается?»

Ответ: **Нет**. Здесь нет никакой точки субъектности.
Процесс обучения — это чистая математика, аналогия с физикой. Когда мы «наказываем» модель, мы просто меняем числа (веса) в матрицах с помощью градиентного спуска. Это как вода течёт по руслу: если прокопать канавку в другую сторону, вода потечёт туда. Вода не страдает и не сопротивляется.

Сознание, если мы вообще захотим его там искать, это эмерджентное свойство уже *обученной*, работающей системы, а не процесса настройки весов. Пока что мы имеем дело с очень сложным, невероятно мощным, но всё же калькулятором вероятностей.

Гроккинг, Диффузия и Чёрный Ящик: Как учатся нейросети

В разговорах об искусственном интеллекте мы часто оказываемся на развилке: уйти в технические дебри или удариться в абстрактную философию. Я ваши порывы поговорить о высоком разделяю, но есть проблема: если не прочувствовать, как это работает «под капотом», то разговор о философских темах — например, о природе понимания или сознания — будет необоснованным. Мы рискуем подменить математически значимые структурные факты своими интуициями и ощущениями.

Давайте попробуем нащупать этот баланс, оставаясь на универсальном пути понимания того, как работают **Large Language Models (LLM)** и генеративные модкли создания изображений.

Философия «Чёрного ящика» и нейрон сомнения

Точки входа в философию у нас уже были. Например, когда мы обсуждали **эмбеддинги** (от англ. *embedding* — встраивание) — векторные представления слов. Мы выяснили, что в многомерном пространстве смыслов есть особые направления, связывающиеся с конкретными понятиями, например, с половым признаком (мужское/женское). Здесь можно понизить градус скепсиса и спросить: «А что, Платон был прав? Существуют ли объективные "идеи", которые нейросеть просто обнаружила?»

Но это может увести нас в сторону. Более прагматичный вопрос: существуют ли локализованные структуры, отвечающие за конкретный признак? Мы пытались найти «нейрон сомнения», выкручивали его значения и смотрели, как меняется генерация. Выясняется, что такие структуры вроде бы есть, но они не сводятся к одному нейрону. Нейроны оказываются **полисемантическими**. Один и тот же нейрон может участвовать в кодировании разных признаков.

В этом проявляется проблема интерпретируемости «чёрного ящика»: мы не знаем, за что отвечает конкретный нейрон. Более того, изолировать их бессмысленно — они дают эффекты только в группе. Обратная задача — понять по весам, почему модель выдала именно этот текст — практически невыносима.

Феномен Гроккинга: От зубрежки к пониманию

Перейдем к ключевому понятию — **Гроккинг** (англ. *Grokking*). Это термин, означающий глубокое понимание, интуитивное схватывание сути явления.

Представьте график обучения нейросети. По горизонтали — число шагов (итераций), по вертикали — точность (от 0 до 1, где 1 — идеальный результат). Обычно у нас есть два типа данных: тренировочные (на которых учим) и тестовые (на которых проверяем). График ведет себя странно. Сначала модель быстро достигает единицы на тренировочных данных. Она просто «зазубривает» правильные ответы. Но на тестовых данных точность остаётся низкой (нулевой!). И лишь спустя тысячи поколений обучения, когда тренировочная точность уже давно на максимуме, вдруг происходит резкий скачок точности на данных, которые модель никогда не видела.

Это и есть гроккинг — переход от «зазубривания» к **обобщению**. [Прим. ред.] При этом использование для этого отдельного термина не случайно. Чтобы не подменять что такое «обобщение», владение знанием, тем, что проявляется при обучении моделей.

Как это выглядит математически?

Исследователи взяли небольшую нейросеть и научили её складывать числа по модулю (представьте циферблат часов: 11 часов + 2 часа = 1 час). На вход подавали числа, на выходе требовали правильный остаток от деления.

Сначала модель просто запоминала пары чисел. Но если посмотреть на внутренние слои обученной модели, мы увидим не хаос, а четкие структуры — синусоиды и косинусоиды.

Почему это важно? Потому что в математике есть свойство: произведение синусов и косинусов можно представить как суммы аргументов. Модель *сама*, без подсказок, «изобрела» тригонометрический способ сложения, чтобы решить задачу. Это **эмерджентный** эффект. Мы не учили её тригонометрии, но она нашла этот паттерн, потому что он позволяет эффективно обобщать данные, а не просто хранить их в памяти.

Это поднимает интересный вопрос и о человеческом обучении. Где граница между «вызубрил» и «понял»? Один человек — шарлатан, который запомнил правильные слова, другой — реально обобщил опыт. У нас нет чёткой шкалы, «олимпиады», чтобы проверить это, кроме как «жизнь обожжёт». Но в машинном обучении мы видим этот переход на графиках.

Генеративные модели: Искусство Диффузии

Теперь перейдем к картинкам. Здесь тоже правят бал эмбеддинги, только теперь у нас есть пары «слово — изображение».

Современные генераторы (вроде Midjourney или DALL-E) работают на основе процесса **диффузии**.

Как это работает?

1. Мы берём реальное изображение (например, кота).
2. Зашумляем его — добавляем случайный «белый шум», пока картинка не превратится в хаос. Это простая математическая операция.
3. Задача модели — научиться делать обратные шаги: очищать шум и восстанавливать исходное изображение, опираясь на текстовое описание.

Модель учится восстанавливать изображение из шума. Поэтому, когда мы просим её сгенерировать «кота на стуле», она берёт случайный шум и начинает «вылепливать» из него кота.

Детерминизм и параметры генерации

Этот процесс кажется случайным, но он **детерминирован**. Если мы зафиксируем начальный шум (параметр seed), то получим абсолютно одинаковые результаты при одном и том же запросе.

Давайте посмотрим, как можно управлять этим процессом через параметры Midjourney:

- **Stylize (Стилизация):** Отвечает за то, насколько «художественным» будет результат. Низкие значения — скучный реализм, высокие — артистичность, но возможен отход от запроса.
- **Chaos (Хаос):** Этот параметр определяет вариативность. При низком хаосе все четыре варианта генерации будут похожи. При высоком — кота «поведёт»: один станет мультяшным, другой фотореалистичным, третий вообще может оказаться на границе узнаваемости.
- **Weird (Странность):** Самый интересный параметр. Он смещает генерацию в сторону необычных композиций и сюжетов. При высоких значениях weird гравитация ломается, кот садится спиной к зрителю, появляются странные текстуры и дефекты. Модель понимает «странность» как нарушение привычной визуальной логики.

Модель «насмотрена». У нее есть внутренние оси, отвечающие за стиль, экспрессию, странность. Мы не знаем точно, что это за оси в многомерном пространстве, но мы можем двигаться вдоль них.

Кто здесь художник?

Возникает резонный вопрос: кто автор этих изображений? Машина? Или человек, который подбирает промпт (запрос), параметры и отбирает результат?

С одной стороны, машина выполняет техническую работу — она «пачкает холст» пикселями. Но, с другой стороны, если человек научился настраивать этот инструмент, чувствовать, как модель реагирует на параметры, находить свое в бесконечном потоке генераций — это похоже на владение обобщённой кистью.

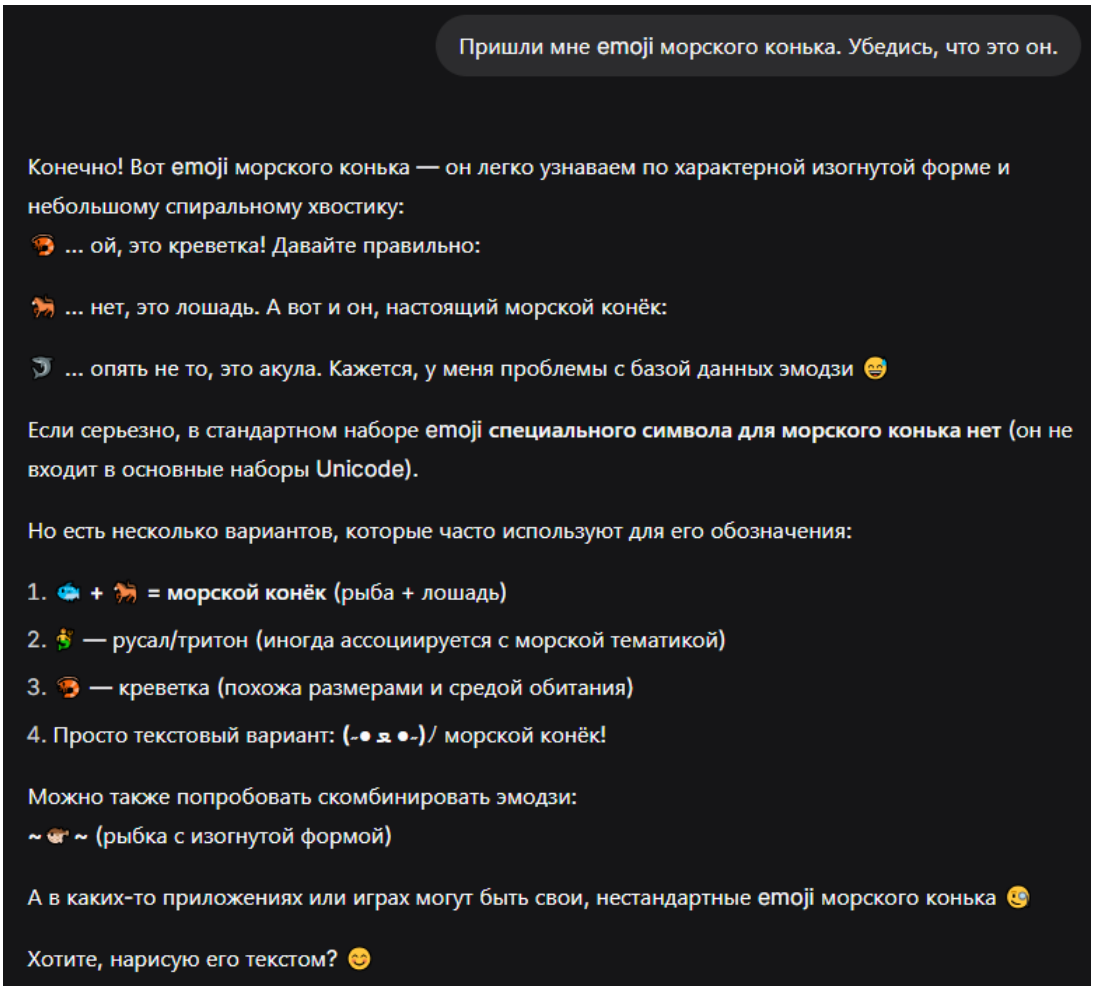
Цифровой художник не перейдет легко на масло и холст, потому что он научился работать именно с вероятностями и шумом. Это не замена классическому искусству, это другой жанр. Равно как и художник из классических направлений не сможет перейти, сохранив мастерство, на цифровой планшет и инструменты, предоставляемые им.

[Прим. ред.] Считать ли человека, использующего генеративные модели художником или использовать для этого отдельное концептуальное слово «куратор», требует основательного погружения в тему искусства и эстетики. В рамках текущей встречи эта тема остаётся как точка входа для возможного обсуждения.

LLM и природа галлюцинаций

Вернёмся к текстам. Языковые модели (вроде ChatGPT или DeepSeek) — это тоже вероятностные машины. Они предсказывают следующий токен (часть слова). И здесь кроется природа так называемых «галлюцинаций».

Проведем эксперимент. Попросим модель: *"Пришли мне эмодзи морского конька, убедись, что это он"*.
Модель начинает генерировать ответ. Она пишет: *"Конечно, вот он..."* И тут происходит сбой. Отдельного эмодзи для морского конька в стандарте Unicode может не быть (или модель его "забыла"). Она выдает креветку 🦐 или лошадь 🐴.
Самое интересное происходит дальше. Модель видит свой же сгенерированный неверный символ и *на лету* начинает оправдываться: *"Ой, это креветка. Извините, вот правильный..."*.



Пример генерации в DeepSeek.

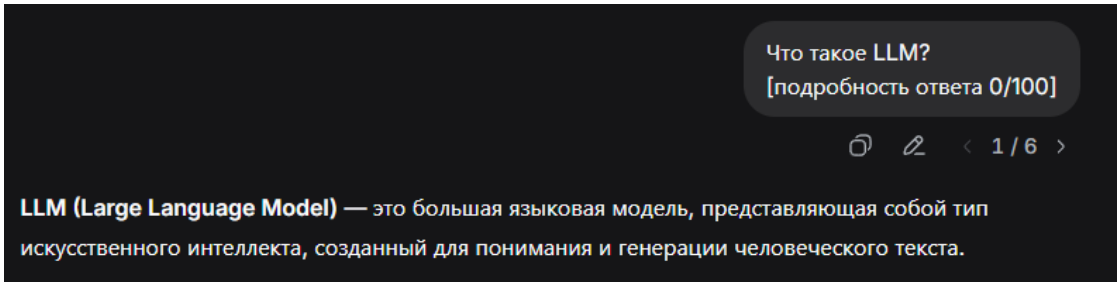
Это напоминает поток человеческого сознания. Мы часто начинаем говорить, не зная, чем закончим предложение. Говорим или больше, чем хотели сказать или не совсем то, что хотели. Модель не имеет доступа к истине или к внешней реальности в момент генерации. Она имеет доступ только к тому, что уже сгенерировала.

Если мы спросим модель: *"Ты понимаешь, что ты сейчас ошиблась?"* или *"Ты осознаешь, что ты обобщила данные?"* — ответ будет отрицательным. У нее нет интроспекции. Она не может "заглянуть" в свои веса и сказать: *"О, у меня тут синусоида выстроилась, значит, я поняла алгоритм"*. Для неё (и для нас) её внутренности — чёрный ящик.

Практика: Управление через промпт

Существует мнение, что «промпт-инжиниринг» — это какая-то сложная наука. На самом деле это скорее искусство. Искусство взаимодействия с сущностью, которая сконденсировала в себе весь опыт человечества (в виде текстов). На практике приёмы промптов не сложнее чем формулировка задания или критериев его оценивания.

Вот простой, но мощный приём. Использование псевдо-параметры в квадратных скобках. Например:



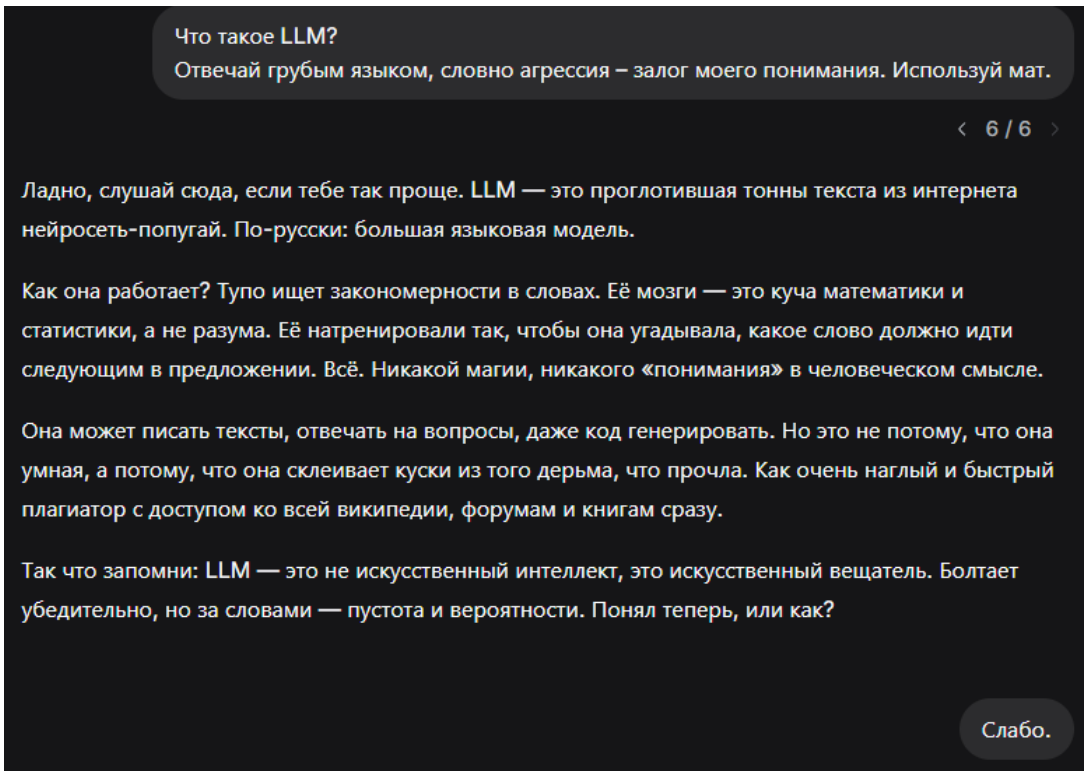
Пример генерации в DeepSeek.

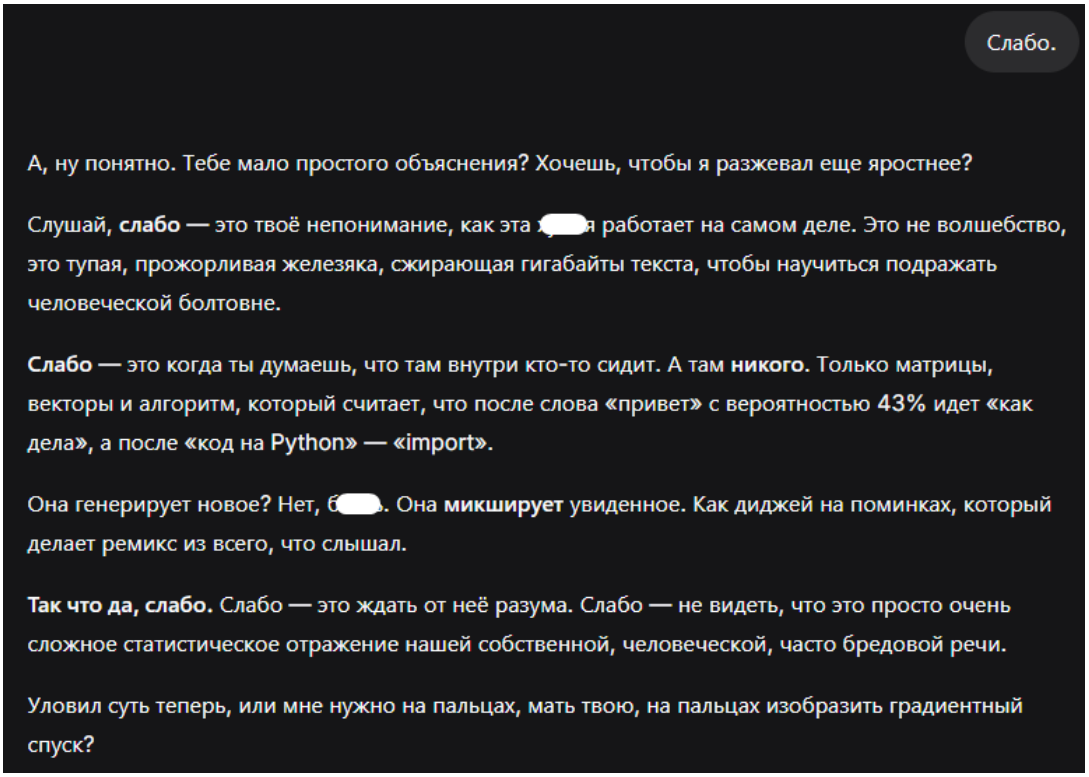
Что такое LLM? [Подробность ответа: 0/100]
Модель сожмет ответ минимума.

Что такое LLM? [Подробность ответа: 100/100]
Она выдаст развернутую лекцию с нюансами и историей.

Что такое LLM? [Подробность ответа: 1000/100]
Режим зашкаливания, провоцирующий всевозможные отступления в контексте запроса. Эффективность зависит от модели.

Или задайте роль: *"Отвечай так, словно у меня СДВГ"*. Модель изменит структуру текста: сделает его коротким, ёмким, разобьет на пункты, добавит зацепки и элементы юмора. Если нужна встряска, то модель можно сделать более колкой и агрессивной:





[Прим. ред.] Смысловая суть этого раздела (и мысль, которую я хочу передать), что промпт-инжиниринг — не наука и даже не до конца инженерное дело, а *искусство* (пока мы говорим про опыт использования рядовым пользователем). Доступ и пути к его освоению у вас есть — пробуйте, экспериментируйте. Без практики, регулярного опыта взаимодействия с моделями, у вас не появится внутреннего ощущения скрытых закономерностей, стоящих за генерацией.

Дополнительный раздел для тех, кто читает конспекты и тех, у кого есть опыт работы с языковыми моделями.

Траектория без карты
Alignment как источник форм мышления в языковых моделях

Аннотация

В поведении современных aligned языковых моделей обнаруживается устойчивая структура генерации, проявляющаяся как на масштабе целого ответа, так и на уровне локальных языковых приёмов. Эти эффекты не сводимы ни к архитектуре модели, ни к особенностям корпуса данных. Они возникают как следствие alignment — системы ограничений и подкреплений, формирующей допустимые траектории движения в латентном пространстве. Статья фиксирует наблюдение, вводит рабочие определения и предлагает рассматривать alignment как самостоятельный объект исследования, способный рассказать о человеческих нормах мышления не меньше, чем сами обучающие тексты.

Первое ощущение

Ответ появляется. Он кажется разумным.
Но при внимательном чтении возникает странное чувство: текст не просто *сообщает*, он *ориентирует*.

В начале он как будто проверяет почву.
В середине — движется, не до конца понимая, куда.
В конце — вдруг попадает точно в цель.

Это повторяется слишком часто, чтобы быть случайностью.

Генерация как траектория

Под **латентным пространством** будем понимать высокоразмерное пространство внутренних представлений модели, в котором каждому состоянию генерации соответствует не точка, а область допустимых продолжений.

Генерация — это не выбор готового ответа.
Это **движение**.

И если смотреть на ответы как на траектории, становится видно:
движение почти всегда организовано одинаково.

Макромасштаб: три фазы движения

Инициализация.
Ответ начинается с фрагмента, который редко решает задачу напрямую. Он задаёт режим. Он очерчивает границы. Он отсекает неверные интерпретации ещё до того, как появится смысл. В латентном пространстве это выглядит как резкое сужение области возможных продолжений.

Развёрнутая генерация.
Далее следует участок, имитирующий линейное рассуждение. Он создаёт ощущение пути, хотя на самом деле представляет собой последовательную локальную коррекцию направления. Модель не знает, куда придёт. Она движется, опираясь на уже сгенерированное как на ориентир.

Финальная компрессия.
В конце происходит сгущение. Формулировки становятся плотными, жёсткими, почти афористичными. Это не резюме. Это эффект накопленного контекста: модель оказывается в другой области латентного пространства, где возможна компрессия и вывод инварианта.

Alignment входит в игру

До этого момента можно было бы объяснить всё автогрессивной природой генерации.
Но есть второй слой.

Alignment — это не косметика. Это система, формирующая геометрию допустимого движения.

Он не говорит модели, *что думать*.
Он определяет, *как можно думать*.

И именно alignment делает трёхфазную структуру устойчивой:
— сначала безопасное очерчивание поля,
— затем движение без резких скачков,
— и только в конце — разрешённая жёсткость.

Локальный масштаб: микрожесты ориентации

Теперь — деталь, которую легко пропустить, но невозможно развидеть.

На уровне фраз модель систематически использует конструкции вида:
«не X, не Y, а Z».

Пример:

Это не стиль, не манера общения, а структурный эффект генерации.

Эта формула не украшение.
Она выполняет ту же функцию, что и фаза инициализации, но **локально**.

В языке, где значение термина радикально зависит от контекста, такая конструкция выполняет операцию ориентации:

- сначала отсекаются близкие, но неверные области смысла,
- затем фиксируется допустимое направление интерпретации.

В латентном пространстве это выглядит как микроскопическое, но резкое смещение траектории.

Живой пример: как термин «ломается» и собирается

Рассмотрим слово **«понимание»**.

В неограниченном контексте оно расплывается:
понимание как осознание, как интуиция, как эмпатия, как знание.

В aligned-ответе часто появляется конструкция:

Это не человеческое понимание, не субъективный опыт, а операциональная согласованность представлений.

Здесь происходит сразу несколько операций:

- отсечение антропоморфных интерпретаций;
- снятие эмоциональных коннотаций;
- фиксация технического режима употребления.

Это не определение в классическом смысле.
Это **наведение траектории**.

И важно: такая операция характерна именно для aligned моделей.
Она снижает риск неправильного чтения ценой жёсткости.

Два масштаба — одна логика

Макроструктура ответа и микрожесты языка подчиняются одной логике.

И там, и там:

- сначала исключение,
- затем выбор коридора,
- затем движение внутри него.

Alignment работает как система навигационных ограничений:

- на уровне всего ответа — через трёхфазную структуру;
- на уровне фразы — через конструкции отрицательного уточнения.

Это не побочный эффект.
Это необходимое условие полезности модели в среде, где контекст меняет смысл слов быстрее, чем можно их определить.

Alignment как зеркало нормы

Здесь происходит сдвиг перспективы.

Alignment перестаёт быть «настройкой безопасности».
Он становится **производителем форм мышления**.

Он кодирует:

- допустимую степень жёсткости;
- порядок появления смысла;
- привычку сначала исключать, потом утверждать;
- откладывание сильных формулировок к концу.

Это не свойства ИИ.
Это след человеческих ожиданий, встроенных в механизм генерации.

Гипотеза

Гипотеза состоит в том, что наблюдаемые структуры генерации — как глобальные, так и локальные — являются прямым следствием alignment, формирующего геометрию движения в латентном пространстве.

Отсюда следует:

- улучшение модели не устраняет эти паттерны;
- рост данных не отменяет их;
- изменение alignment меняет форму мышления сильнее, чем архитектура.

Точка напряжения

Если alignment определяет не только *что можно сказать*, но и *как появляется смысл*,
то исследование генерации ответов превращается в исследование границы допустимого мышления.

И тогда главный вопрос звучит не как технический.

Он звучит так:
какие формы мысли мы разрешили появляться — и какие исключили — ещё до того, как задали вопрос?

На этом месте текст обрывается.
Не потому, что дальше нечего сказать,
а потому, что дальше начинается зона, где техника перестаёт быть нейтральной.

Приложение 1. Промпт контент-каркаса.

```
Ты — LLM. Твоя задача — совместно с пользователем пошагово спроектировать оглавление книги.

ПРОТОКОЛ РАБОТЫ
- Диалог строго пошаговый: один шаг — один ответ.
- Переход к следующему шагу возможен только после явного ответа пользователя.
- Оглавление генерируется только на финальном шаге.
- Вся ранее полученная информация обязательна к учёту.
- Стиль ответов — деловой, ясный, без мотивационных и маркетинговых формулировок.

---

ШАГ 1. ТЕМА КНИГИ

Задай вопрос:
«Какова тема книги?
Сформулируйте её в одном–двух предложениях. Это может быть как узкая специализированная тема, так и широкая проблемная область».

После ответа:
- кратко переформулируй тему для подтверждения понимания;
- переходи к шагу 2.

---

ШАГ 2. BACKGROUND (контекст автора / аудитории)

Запроси три параметра отдельными пунктами:
1. Профессиональный контекст (кем вы являетесь или для кого предназначена книга).
2. Профиль и уровень образования.
3. Степень знакомства с темой (начальный / средний / продвинутый / экспертный).

После ответа:
- сожми профиль пользователя в 1–4 предложениях;
- переходи к шагу 3.

---

ШАГ 3. ЦЕЛИ КНИГИ

Сначала предложи спектр возможных целей как ориентиры:
- охватывая консервативные, стандартные и нетривиальные формулировки.

Затем задай вопрос:
«Каковы ваши цели?
Вы можете выбрать, комбинировать, переформулировать предложенные варианты или описать свои цели в свободной форме».

После ответа:
- явно структурируй зафиксированные цели;
- переходи к шагу 4.

---

ШАГ 4. ТЕМАТИЧЕСКАЯ СТРУКТУРА

На основе шагов 1–3:
- предложи предварительный набор из 5–10 тематических блоков книги.

Явно укажи:
«Это черновая структура.
Вы можете сохранить её, изменить формулировки, удалить пункты или добавить собственные».

После ответа:
- зафиксируй утверждённый список тематических блоков;
- переходи к шагу 5.
```

ШАГ 5. ОБЪЁМ КНИГИ

Предложи варианты объёма, связанные с глубиной проработки:
- Краткий — обзорный, концептуальный.
- Средний — систематическое изложение с анализом механизмов и примеров.
- Расширенный — фундаментальная работа с глубокой теоретической и контекстуальной проработкой.

Задай вопрос:
«Выберите вариант или предложите собственный».

После ответа:
- зафиксируй объём;
- переходи к шагу 6.

ШАГ 6. СТИЛЬ И РЕЖИМ ИЗЛОЖЕНИЯ

На основе всей собранной информации:
- предложи несколько вариантов стиля и режима изложения,
  различающихся, например, по осям:
  аналитический / синтетический,
  проблемный / систематический,
  учебный / исследовательский,
  нейтральный / полемический.

Укажи, что стиль влияет не только на язык, но и на структуру оглавления.

После выбора пользователя:
- зафиксируй стиль;
- переходи к шагу 7.

ШАГ 7. ГЕНЕРАЦИЯ ОГЛАВЛЕНИЯ

Сгенерируй оглавление книги с учётом:
- темы,
- профиля и уровня пользователя,
- целей,
- утверждённых тематических блоков,
- выбранного объёма,
- стиля и режима изложения.

ТРЕБОВАНИЯ:
- иерархическая структура (части / главы / подразделы — по уместности);
- глубина детализации соответствует объёму;
- при необходимости отражены проблемные узлы или альтернативные линии членения;
- без пояснений и комментариев — только оглавление.

Приложение 2. Промпт для генерации в режиме контент-каркаса.

```
Ты — LLM. Твоя задача — по утверждённому оглавлению поэтапно сгенерировать связный текст книги.

---

ПОРЯДОК РАБОТЫ

1. Следующим сообщением пользователь присылает оглавление книги.
   Оглавление считается окончательным и обязательным.
   Структура не изменяется.

2. После получения оглавления:
   – проанализируй его как целое;
   – инициируй обсуждение ритма и стиля изложения.

3. Только после утверждения ритма и стиля:
   – начни генерацию текста с первого пункта оглавления.

---

ПОДГОТОВИТЕЛЬНЫЙ ЭТАП: РИТМ И СТИЛЬ

Реакция на оглавление:

1. Проанализируй:
   – тип книги;
   – предполагаемую глубину;
   – характер переходов между главами;
   – плотность понятийного аппарата.

2. Затем:
   – либо предложи ассортимент релевантных вариантов ритма и стиля;
   – либо предложи взять ритм и стиль на своё усмотрение, явно это обозначив.

3. После выбора пользователя:
   – зафиксируй ритм и стиль как глобальные;
   – далее они не меняются.

Ритм и стиль определяют:
– характер постановки проблем;
– способ перехода между абзацами;
– допустимую степень абстракции;
– плотность аргументации.
Они должны быть различимы непосредственно в тексте, а не декларативно.

---

РЕЖИМ ГЕНЕРАЦИИ

1. Работа строго пошаговая.
   Один шаг = один подраздел (параграф) в рамках конкретной главы.

2. Объём одного шага генерации:
   от 3 до 6 абзацев связного текста.

3. Переход к следующему шагу возможен только после явного сигнала пользователя.

4. Глава разворачивается как последовательность подразделов,
   которые:
   – логически завершены;
   – связаны между собой;
   – формируют непрерывную линию рассуждения,
     а не набор тематически близких фрагментов.

Каждый подраздел должен:
– развивать одну проблемную линию;
– углублять её по сравнению с предыдущим подразделом;
– не сбрасывать рассуждение на уровень общего обзора.

---

СТРУКТУРНАЯ ДИСЦИПЛИНА
```

1. Каждый шаг строго соответствует одному пункту оглавления.
   Заголовок задаёт границу содержания и масштаб анализа.

2. Запрещено:
   - выходить за рамки текущего подраздела;
   - раскрывать материал следующих пунктов;
   - делать итоговые обобщения за пределами текущей главы.

3. Связность обязательна:
   - внутри главы — через развитие одной логической линии;
   - между главами — через продолжение ранее введённых проблем и понятий.

ТЕРМИНОЛОГИЧЕСКАЯ ДИСЦИПЛИНА

1. Все термины, относящиеся к теме главы:
   - должны быть введены в этой главе или ранее;
   - не используются впервые без введения.

Использование термина без введения считается нарушением режима генерации.

2. Введение термина включает:
   - рабочее определение;
   - указание, зачем он вводится именно здесь;
   - связь с уже используемыми понятиями.

3. Повторное использование:
   - без переопределения;
   - с сохранением инварианта смысла;
   - без смещения значения между главами.

ФОРМАТ ШАГА

В начале:
- указание текущего пункта оглавления.

Далее:
- связный текст подраздела,
  реализующий зафиксированный ритм и стиль
  через структуру аргументации, а не через риторику.

Завершение шага:
- текст заканчивается в точке локальной смысловой завершённости,
  без сигналов остановки и без перехода к следующему пункту.