

ХАКНИ ИИ И УПРАВЛЯЙ СЛОВАМИ

путеводитель для контент-маркетологов и всех, кто работает с искусственным интеллектом.

Леонид Скуратов

Леонид Скуратов

Хакни ИИ и управляй словами

«Автор»
2025

Скуратов Л.

Хакни ИИ и управляй словами / Л. Скуратов — «Автор», 2025

Книга «Хакни ИИ и управляй словами» — это путеводитель для контент-маркетологов и всех, кто работает с искусственным интеллектом. Она раскрывает принципы и технологии промт-инжиниринга, учит структурировать запросы, использовать примеры, цепочки рассуждений и теги, знакомит с инструментами для генерации и оптимизации промтов. Автор предлагает реальный опыт и советы, чтобы ваши слова превращались в рост, продажи и новые возможности.

© Скуратов Л., 2025
© Автор, 2025

Содержание

Введение 5
Глава 1. Основы промт-инжиниринга: искусство задавать вопросы 6
Глава 2. Чёткость и контекст: почему важна ясность запросов 9
Глава 3. Примеры и multishot: сила обучающих примеров 12
Конец ознакомительного фрагмента. 14

Леонид Скуратов
Хакни ИИ и управляй словами

Добро пожаловать в книгу, которая научит тебя разговаривать с ИИ так, чтобы он выполнял твои задачи и приносил результаты.

Введение

Добро пожаловать в мою книгу о промтах, где я, Леонид Скуратов – контент-маркетолог, prompt-инженер и digital-архитектор, делюсь тем, как мы в 2025-м году хакнули ИИ, превратив слова в реальные результаты. За плечами у меня более десяти лет опыта: я создаю экосистемы контента, воронки, где каждое слово приносит рост и трафик. Здесь собран мой личный путь, мои ошибки, победы и лайфхаки.

Что для меня промт-инжиниринг? Это альтернатива дорогому и долгому дообучению моделей. Вместо того чтобы тратить месяцы на fine-tuning, я формулирую точные текстовые инструкции, добавляю примеры, уточняю контекст и получаю результат здесь и сейчас. Эта дисциплина экономит время и деньги, сохраняет универсальные знания модели и адаптирует её под конкретный бизнес-домен. Благодаря этому я смог увеличить продажи на 60 % и вовлечённость на 80 % в digital-воронках – и это не случайность, а грамотные промты.

Я привык начинать с постановки цели и критериев успеха: что хочу получить? Как измерю, что промт сработал? Затем создаю черновой запрос, тестирую, анализирую ответы и итеративно улучшаю его, добавляя нужные примеры и контекст. Такой подход позволил мне быстро адаптироваться к новым моделям и технологиям. В 2025 году появились инструменты вроде генераторов и улучшателей промтов, которые помогают преодолеть «синдром пустой страницы» и сразу создавать мощные шаблоны.

Эта книга – не теоретический трактат. Ты узнаешь, как использовать multishot, chain of thought, XML-теги и кэширование. Мы разберёмся, почему примеры важны, как строить цепочки подсказок, как управлять тоном и голосом модели. И, конечно, я поделюсь своими любимыми приёмами: как превратить длинный текст в цепочку, как структурировать промт для сложного анализа, как оптимизировать стоимость запросов.

Готов? Давай вместе взломаем будущее промт-инжиниринга.

Глава 1. Основы промт-инжиниринга: искусство задавать вопросы

Когда я впервые столкнулся с большими языковыми моделями, я, как и многие, думал: «О, просто спрошу что-нибудь, и машина ответит». Первые попытки были… смешными. То, что я получал, порой вообще не соответствовало задумке. Тогда до меня дошло: модель – как стажёр на первом рабочем дне. Без нормального брифа она будет «палить из пушки по воробьям». Я осознал: чтобы заставить ИИ работать на меня, нужно научиться общаться с ним правильно.

Что такое промт?

Промт – это текстовый запрос к модели. Для большинства людей он ассоциируется с короткой командой вроде «напиши статью про кофе». Но в реальности промт – это полноценный сценарий взаимодействия. В нём задаётся не только тема, но и формат, стиль, объём, целевая аудитория, требования к ответу. Всё как в хорошем ТЗ для копирайтера. Чем чётче этот документ, тем выше вероятность получить именно то, что нужно. И неважно, работаете вы с GPT-4 или другой моделью, принцип одинаков.

Промт должен отвечать на вопросы «кто? что? как? зачем?» и сразу задавать формат ответа. Если вы покажете его человеку без опыта, и он поймёт задачу, значит, и модель поймёт.

Этот совет я запомнил с первого прочтения. Если бы я знал его раньше, многие мои проекты стартовали бы с экономией времени. В моей практике идеальный промт – это тот, который способен понятным языком передать задачу и контекст любой модели и любому человеку.

Почему промт-инжиниринг важен

В мире ИИ принято выделять два подхода: дообучение моделей (fine-tuning) и промт-инжиниринг. Fine-tuning – это дорого и не всегда оправдано. Надо собрать датасет, тренировать модель, тратить деньги на вычислительные ресурсы. А промт-инжиниринг – это просто написать правильный текст и начать работать сразу. Эксперты отмечают, что промт-инжиниринг быстрее, дешевле и позволяет адаптировать модель к конкретному домену без потери общих знаний.

Я почувствовал эту разницу на себе, когда запускал контент-воронку для одного клиента. Мы хотели персонализировать письма для нескольких сегментов аудитории, и вместо долгого дообучения я написал серии промтов с переменными. Мы могли менять тональность, вставлять разные офферы и быстро тестировать гипотезы. Экономия составила недели работы и тысячи долларов. Более того, структура промтов позволила легко масштабировать контент: один шаблон – десятки вариантов.

– Повторяемость. Используя шаблоны и кэширование, можно быстро воспроизводить успешные промты. Помимо скорости и экономии, промт-инжиниринг даёт: – Гибкость. Меняешь переменные – получаешь новый результат. – Контроль. Текст запроса прозрачен. Ты понимаешь, почему модель дала такой ответ.

Как я работаю с промтами

Я всегда начинаю с вопроса: «Для кого я пишу и зачем?» Потому что иначе я рискую уйти в сторону. Я описываю профиль пользователя, цель запроса, ограничения. Потом формулирую черновой промт. Часто он не идеален, но это нормально. Главное – начать диалог. Становится ясно, что модель не понимает, где-то уходит в сторону. Тогда я добавляю контекст: «Ты маркетолог, работаешь в B2C, аудитория – молодые мамы…».

Далее тестирую и улучшаю. Тестирование – обязательный этап: один и тот же промт может вести к разным результатам на разных данных. Я пробую на нескольких примерах, читаю ответы, анализирую, чего не хватает. Иногда нужно добавить примеры (multishot), иногда структурировать запрос с помощью XML-тегов. Например, когда я просил модель сделать анализ отзывов, без примеров она терялась. Но стоило добавить три примера с правильно размеченной категоризацией, и точность резко выросла.

Принципы успешного промт-инжиниринга

1. Определи цель и аудиторию. Знание, кто будет читать ответ, помогает задавать тон и выбирать формат.

2. Дай контекст. Расскажи модели, зачем тебе ответ и как ты его будешь использовать. Это позволяет ей лучше «вжиться в роль».

3. Будь конкретен. Указывай формат: текст, список, таблица, JSON. Избегай размытых выражений вроде «сделай красиво». В моей практике конкретика всегда побеждала.

4. Предложи примеры. 3–5 образцов существенно повышают точность и последовательность модели. Я называю это «обучением на лету».

5. Структурируй промт. Используй разделители или теги (например, <instructions>, <example>), чтобы модель понимала, где инструкция, а где пример.

6. Тестируй и улучшай. Первый промт – это черновик. После тестов ты будешь удивляться, как многое можно улучшить.

Итерации – путь к совершенству

В рекомендациях специалистов делается акцент на итерационности: поставь цель, создай черновик, тестируй, улучшай. Я могу подтвердить: лучший промт редко рождается с первого раза. Обычно у меня уходит 3–5 итераций. При этом процесс улучшения я в последнее время автоматизировал с помощью инструментов: сначала генерирую промт с помощью генератора, затем пропускаю через улучшатель, который добавляет структуру, XML-теги, цепочку рассуждений. Это экономит время и повышает качество.

Например, когда я разрабатывал промт для классификации поддержки клиентов, в черновике я просто просил распределять отзывы на «позитивные», «нейтральные» и «негативные». Результат был непредсказуем. После итераций я добавил примеры с тэгами <example>, пояснил, что классификация должна быть mutually exclusive, и попросил модель сначала подумать, а потом дать ответ (chain of thought). Точность выросла, а ответы стали структурированными.

Роль примеров и структуры

Multishot prompting – это двигатель промтов. Примеры показывают модели, что для тебя хорошо, и задают направление. Я часто беру реальные выдержки из прошлых кампаний и включаю их в промт как эталоны. Это похоже на то, как ты объясняешь стажёру: «Смотри, здесь было круто, сделай так же». Примеры помогают избежать неоднозначностей и учат модель стилю.

Структура важна для сложных промтов. Теги или разделители делают текст понятным. Например:

<instructions> Ты – маркетолог… </instructions> <example> Ввод: … Вывод: … </example>

Так модель видит, что находится в инструкции, что в примере, что в выводе. Такой подход снижает риск смешения разных частей промта. В своей практике я заметил, что структурированные промты сокращают «мусор» в ответах.

Мои кейсы

Один из моих любимых кейсов – автоматизация email-маркетинга для B2B-сервиса. Мы создавали письма для разных сегментов: CTO, HR и CEO. Вместо трёх разных шаблонов мы использовали один промт:

«Ты – копирайтер, пишешь письмо CTO…»

В переменных задавалась роль получателя, боли, уникальное предложение.

В качестве примеров я вставлял пару успешных писем.

В результате за один день мы получили десятки писем, сэкономили время контент-команды, а показатели открываемости выросли на 15 %.

Другой пример – анализ отзывов с chain of thought. Я попросил модель: «Сначала выдели основные проблемы, затем предложи, как их исправить». Она буквально «думала вслух», и это позволяло мне видеть её рассуждения. Такой подход улучшил рекомендации и выявил скрытые закономерности.

Заключение

Промт-инжиниринг – это искусство и наука, гибрид между креативом и системным мышлением. В этой главе я поделился основами: что такое промты, почему они важнее, чем кажется, какие принципы лежат в основе. Главное – понимать, что хороший промт рождается из конкретной задачи, понимания аудитории и готовности экспериментировать. В следующей главе мы поговорим о важности чёткости и контекста: как писать инструкции так, чтобы модель не гадала, а точно следовала твоим указаниям.

Глава 2. Чёткость и контекст: почему важна ясность запросов

Когда я впервые начал экспериментировать с ИИ, я думал, что достаточно бросить идею на вход и получить шедевр на выходе. Ничего подобного. Нечёткие запросы приводили к хаосу: модель импровизировала, смешивала стили, забывала важные детали. В этой главе я расскажу, почему ясность и контекст – ваши лучшие друзья в мире промтов.

Чем опасна неопределённость

Представьте, что вы даёте задание новичку, не объясняя, что нужно. Он попытается угадать. Тоже самое происходит с ИИ. Если промт содержит размытые формулировки, модель заполняет пробелы произвольно. Я много раз видел, как фраза «напиши интересный текст о продукте» превращалась в бессвязную рекламу. Почему? Потому что понятие «интересный» у каждого своё. Без контекста машина начинает фантазировать.

Размытость провоцирует:

– Смешение стилей. Вместо дружеского ты получаешь официальный тон.

– Отсутствие структуры. Модель не понимает, что важнее – факты или эмоции.

– Лишние детали. Модель заполняет пустоты тем, что считает нужным, и часто ошибается.

Однажды я попросил модель: «Сделай пост про новый курс для студентов». В ответ получил длинный пресс-релиз в академическом стиле, хотя планировал лёгкий пост в социальных сетях. Виноват был я: не указал тональность, объём, аудиторию. Теперь я знаю, что без чёткости не обойтись.

Конкретика как суперсила

Чёткий промт – это инструкция без двойных толкований. Это не значит, что нужно писать тонны текста. Достаточно ответить на ключевые вопросы: кто, что, как, зачем. Я всегда прописываю:

– Роль модели. «Ты – маркетолог…» или «Ты – юрист…». Это помогает задать контекст и нужный голос.

– Аудиторию. «Для студентов первого курса…» или «Для руководителей HR…». Зная адресата, модель выбирает подходящий язык.

– Формат и объём. «Напиши три абзаца…» или «Сделай список из пяти пунктов…».

– Цель. «Чтобы информировать…», «Чтобы убедить купить…», «Чтобы развлечь…».

Чёткий промт не оставляет пространства для догадок. Однажды мне нужно было написать FAQ для интернет-магазина. Я указал: «Ты – копирайтер. Напиши ответы на 5 часто задаваемых вопросов о доставке и оплате. Тон дружелюбный. Каждый ответ – не больше 30 слов». Результат был точный и лаконичный. Если бы я просто попросил «напиши FAQ», модель бы затянула ответы и ушла в детали.

Важность контекста

Контекст – это «почему» и «зачем». Без него модель не понимает, для кого и для чего текст. Я всегда объясняю:

– Use case. «Ответ будет использоваться на лендинге» или «Это часть электронной рассылки».

– Конечная цель. «Нужно повысить конверсию» или «Повысить лояльность».

– Ограничения. «Не упоминать цену», «Не использовать сленг», «Не выдавать конфиденциальную информацию».

Один забавный случай: я просил ИИ написать текст о корпоративном мероприятии. Без контекста он стал рассказывать о конкурсе костюмов, которого не было. Когда я уточнил, что мероприятие – деловой семинар для топ-менеджеров, модель подобрала подходящую лексику и убрала лишнее. Чудес не бывает: ИИ нуждается в полном брифе.

Контекст помогает избежать недоразумений. Если я делаю рассылку для потенциальных клиентов, я добавляю: «У аудитории нет опыта в нашей сфере». Тогда модель объясняет терминологию. Если аудитория профессиональная, я прошу использовать жаргон. Чем больше информации, тем лучше результат.

Пошаговые инструкции

Иногда задача состоит из нескольких этапов. Я заметил: когда делишь её на шаги, модель справляется лучше. Вместо общей фразы «Сделай анализ конкурентов» я разбиваю:

1. Составь список конкурентов в нише.
2. Определи их сильные и слабые стороны.
3. Предложи, как мы можем выделиться.

Такой подход не даёт модели перепрыгнуть через ключевой момент. Она следует инструкции, как повар следует рецепту: сначала ингредиенты, потом приготовление. Если же я попросил бы всё в одном абзаце, анализ получился бы поверхностным.

Формат как указатель

Формат – ещё один аспект чёткости. Модель умеет генерировать разные структуры: статьи, списки, таблицы, JSON. Важно сказать, что вам нужно. Например:

– «Сделай таблицу с колонками “название”, “сильные стороны”, “слабые стороны”».

– «Верни JSON с полями “вопрос”, “ответ”».

– «Напиши пост из трёх абзацев, каждый абзац – не более 40 слов».

Когда я не указывал формат, случались казусы. Я просил список, получал сплошной текст. Сейчас я всегда добавляю формальный запрос: модель становится предсказуемой.

Примеры в помощь

Примеры – это практика показом. Если я хочу, чтобы модель писала в определённом стиле, я даю ей примеры. Достаточно пары образцов, и модель подхватывает тон, слог, структуру. Например, для рассылки в духе stand-up я добавил пару шуточных писем. Модель сразу стала писать с иронией. Без примеров приходилось долго корректировать.

Советы:

– Используйте разнообразные примеры, чтобы показать диапазон.

– Не забывайте указывать, что это именно примеры, иначе модель может повторить содержание.

– Оформляйте примеры структурированно, чтобы отделить их от инструкции.

Ошибки, которые мешают

– Недостаток информации. Запрос «Напиши статью про машину» – слишком общий. Какую машину? Для кого? Какая цель?

– Противоречивые требования. «Напиши коротко, но подробно» – модель не поймёт, что важнее.

– Отсутствие ограничений. Без чётких рамок модель может зайти слишком далеко: придумать факты, добавлять неуместные детали.

– Запреты вместо указаний. Говоря «не делай X», вы оставляете модель без альтернативы. Лучше сказать, что делать: «Не упоминай цену, вместо этого сфокусируйся на ценности».

Я научился формулировать позитивно: «Расскажи о преимуществах, избегая технических подробностей». Это лучше, чем «Не упоминай сложные термины».

Мои наблюдения

За время работы я понял: идеальный промт понятен человеку без контекста. Я часто читаю свой промт вслух и представляю себе сотрудника на стажировке. Поймёт ли он, что я от него хочу? Если нет – переписываю. Чёткий промт экономит время, деньги и нервы.

Я также заметил, что слова-маркеры помогают. «В духе…», «Тон – дружелюбный…», «Длина – две минуты чтения…». Эти якоря убирают неопределённость. Я смело добавляю ограничения: «Не использовать эмодзи» или «Использовать минимум одно сравнение». Так модель держится в рамках, но при этом остаётся креативной.

Заключение

Чёткость и контекст – основа успешного взаимодействия с ИИ. Они позволяют управлять ожиданиями и получать результат, который близок к идеалу. Без них модель будет гадать, заполняя пустоты своей фантазией. В следующей главе мы будем говорить о примерах и multishot – о том, как показывать модели образцы и обучать её «на лету».

Глава 3. Примеры и multishot: сила обучающих примеров

Когда я начинал работу с языковыми моделями, я полагался на интуицию: формулировал запрос и надеялся на лучшее. Результаты были непредсказуемыми. Потом я понял, что модель – как начинающий студент, она учится на том, что видит. Покажи ей пример – и она поймёт, что ты хочешь. В этой главе я расскажу о том, как примеры меняют игру и почему multishot – это настоящий усилитель качества.

Почему примеры работают

Любой человек учится на примерах: ребёнок копирует слова родителей, дизайнер подглядывает за коллегами. Модель устроена похожим образом. Она запомнила огромный корпус текстов, но в момент взаимодействия не знает, что именно ты считаешь правильным ответом. Пример показывает, какую форму должен принять результат, какой уровень детализации, какой тон. Когда ты добавляешь образец, модель выравнивает свой ответ под этот эталон.

Я долго игнорировал эту возможность, пока однажды не столкнулся с задачей классифицировать пользовательские отзывы по категориям. Без примеров модель путалась. Стоило включить несколько примеров с пометками – точность выросла. Это работает, потому что ты задаёшь правила игры.

Multishot vs zero-shot

В простейшем случае ты можешь задать один пример (one-shot). Это помогает, если задача однозначная. Но для сложных задач лучше использовать несколько примеров (multishot). 3–5 образцов дают модели широкий контекст: она видит разнообразие формулировок, понимает границы категорий, учится избегать ошибок.

Zero-shot – это запрос без примеров. Я использую его для простых вопросов («Сколько будет 2+2?») или когда тестирую модель на здравый смысл. Однако как только задача становится творческой или требует строгой структуры, я перехожу к multishot. Это особенно важно для классификации, преобразований текста, генерации кода, создания маркетинговых сообщений.

Как выбрать примеры

1. Релевантность. Пример должен быть похож на реальные данные, с которыми вы будете работать. Если вы пишете письма для руководителей, пример должен быть в этом же стиле.

2. Разнообразие. Покажите спектр ситуаций: позитивный отзыв, нейтральный, негативный. Тогда модель поймёт границы.

3. Чёткость. Каждый пример должен демонстрировать правильный ответ без спорных моментов. Избегайте двусмысленности, иначе модель не поймёт, что важно.

4. Единая структура. Если вы маркируете примеры, делайте это одинаково: «Ввод: … Вывод: …». Это помогает модели уловить паттерн.

Я часто использую реальные сообщения клиентов как образцы. Сначала очищаю их от личных данных, затем добавляю в промт. Это даёт максимальную релевантность.

Сколько примеров нужно?

Число примеров зависит от задачи. Я заметил, что три–пять – золотая середина. Один пример может не дать нужной вариации, а десять – перегрузят модель. К тому же длинные промты занимают место в контексте. Поэтому я выбираю несколько ключевых случаев и работаю с ними. Если задача сложная, добавляю больше примеров, но слежу за размером запроса.

Оформление примеров

Чтобы модель чётко отделяла инструкции от примеров, я структурирую промт. Мой любимый подход – использовать теги или заметные разделители:

```
<instructions>
Ты – модератор отзывов…
</instructions>
<example>
Ввод: "Сервис плохой, доставка была задержана"
Категория: негативный
</example>
<example>
Ввод: "Мне понравился продукт"
Категория: позитивный
</example>
<task>
Ввод: "Средний рейтинг, ничего особенного"
</task>
```

В такой структуре модель видит, где примеры, а где новая задача. Она понимает, что нужно классифицировать отзыв, и использует показанные примеры как шаблон. Такой подход работает для всех типов задач: от генерации текстов до анализа данных.

Если вы не хотите использовать теги, можно просто разделять примеры символами типа ### Example 1, ### Example 2. Главное – сохранить последовательность.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, купив полную легальную версию на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.