

ОСНОВЫ ПРОМТ ИНЖИНИРИНГА



Сабир Алмасов

Сабир Алмасов

Основы промт-инжиниринга

http://www.litres.ru/pages/biblio_book/?art=72581344

SelfPub; 2025

Аннотация

Эта книга – ваш исчерпывающий путеводитель по промт-инжинирингу, который превратит ваши запросы из невнятных просьб в точные, неотразимые команды. Мы проведем вас за руку от создания продающих текстов и фотореалистичных изображений до режиссуры собственных видеороликов и написания для них музыки. Каждая глава – это концентрат рабочих техник, которые превратят ваши самые смелые идеи в готовый продукт. Это практическое пособие, после прочтения которого вы сможете решать любую креативную или рабочую задачу быстрее, качественнее и эффективнее. Инвестируйте в навык, который определит ваше ближайшее десятилетие.

Сабир Алмасов

Основы промт-инжиниринга

Введение!

Давайте начистоту. Вы уже знаете, что такое нейросети. Вы видели, как они создают тексты, рисуют изображения и даже пишут код. Вы, как и миллионы других, используете их в работе или для личных проектов. И скорее всего, вы столкнулись с главной проблемой: результаты часто получаются... средними.

Нейросеть выдает банальный текст, генерирует изображение с шестью пальцами на руке или просто не понимает, чего вы от нее хотите. Вы тратите время на переформулировки, упрощаете запрос и в итоге соглашаетесь на компромисс – на результат, который "в целом сойдет". Знакомая ситуация?

В этом и заключается ключевая разница, которая прямо сейчас определяет, кто останется на обочине технологического прогресса, а кто возглавит его. Разница между *пользователем* и *профессионалом*.

Пользователь тычет в кнопки. Он получает не достаточно хороший результат и считает, что таков предел технологии. Для него нейросеть – это удобный, но капризный инструмент, который иногда экономит пару часов.

Профессионал же понимает, что дело не в инструменте, а в умении им управлять. Он знает: нейросеть – это не просто помощник, это мощнейший ресурс, способный многократно усилить его собственные навыки. Он ставит конкретную задачу. Четко, грамотно и так, чтобы получить предсказуемо превосходный результат с первой попытки.

Этот навык, это умение вести точный и результативный диалог с искусственным интеллектом, и есть **промт-инжиниринг**. И сегодня это не просто полезное умение. Это решающая компетенция.

Эта книга – не сборник "интересных фактов об ИИ". Это концентрированное, практическое руководство по овладению этой компетенцией. Это тренинг, который превратит вас из рядового пользователя в того самого специалиста, за которым сегодня охотятся компании.

Почему это так важно именно сейчас?

Рынок труда меняется с бешеной скоростью. Компании массово внедряют ИИ-решения, инвестируя в них огромные бюджеты. Но при этом они столкнулись с фундаментальным парадоксом: самые передовые и дорогие технологии оказываются практически бесполезными в руках сотрудников, не владеющих методами эффективной работы с ними. В результате, многомиллионные вложения не приносят ожидаемой отдачи, а невероятный потенциал ИИ остается нераскрытым.

Именно поэтому возник колоссальный спрос на новый

тип экспертов. На людей, которые могут стать мостом между бизнес-задачей и возможностями нейросети. Тех, кто способен извлечь из технологии максимум ценности. Тех, кто превращает потенциал ИИ в реальную прибыль и прорывные продукты.

Основная цель этой книги – сделать вас таким специалистом.

Что конкретно вы получите, изучив ее?

Вы научитесь формулировать запросы так, чтобы нейросеть стала вашим послушным и невероятно эффективным исполнителем для любой задачи: от написания сложного аналитического отчета до разработки визуальной концепции нового бренда.

Вы освоите системный подход. У вас будет набор четких методик, техник и протоколов для работы с текстовыми, графическими, видео- и аудио-моделями. Никаких догадок и "метода тыка" – только проверенные стратегии.

Вы получите огромное, неоспоримое преимущество на рынке труда. Пока другие будут указывать в резюме "уверенный пользователь ПК и нейросетей", вы сможете продемонстрировать системный навык, напрямую влияющий на эффективность бизнеса. Это ваш карьерный лифт в любой отрасли.

Кем бы вы ни работали – маркетологом, программистом, дизайнером, менеджером, аналитиком – вы сможете выполнять свою работу в разы быстрее и качественнее. Вы станете

тем сотрудником, который не просто следует трендам, а использует их для достижения выдающихся результатов.

Хватит быть пассивным наблюдателем. Хватит довольствоваться средним уровнем. Эпоха, когда можно было просто "попробовать нейросеть", закончилась. Наступило время, когда нужно уметь работать с ней профессионально.

Эта книга – ваш самый короткий путь к этому мастерству. В ней нет "воды" и философских рассуждений. Только структура, практика и инструменты, которые вы начнете применять уже с первой главы.

Давайте начинать!

Часть I: Фундаментальные принципы промт инжиниринга.

Глава 1. Введение в промт инжиниринг.

Вы наверняка уже работали с нейросетями. И скорее всего, вы замечали разрыв. Разрыв между тем огромным потенциалом, о котором все говорят, и теми средними, порой разочаровывающими результатами, которые вы получаете на практике. Этот разрыв возникает не из-за ограничений технологии. Он возникает из-за отсутствия конкретного навыка – навыка управления этой технологией. Эта книга посвящена именно этому: превращению вашего взаимодействия с искусственным интеллектом в управляемый и предсказуемый процесс. И начнется этот путь с понимания того, как на самом деле работает эта система.

Давайте сразу проясним. Современные генеративные мо-

дели не "думают" и не "понимают" так как это способен делать человек. В их основе лежит сложный математический аппарат для прогнозирования последовательностей. Когда вы даете модели текст, ее основная задача – вычислить и сгенерировать наиболее вероятное продолжение этого текста. Но чтобы сделать это, она должна сперва перевести наш язык в понятный для себя формат. Здесь в игру вступает фундаментальное понятие – **токенизация**.

Нейросеть видит текст не как набор букв или даже слов, а как последовательность **токенов**. Токен – это минимальная единица информации для модели. Им может быть как целое короткое слово, например, «дом» или «и», так и часть более длинного слова или отдельный знак препинания.

Перед началом любой работы модель переводит ваш запрос на свой внутренний, математический язык. Для этого она разбивает текст на эти самые токены и каждому из них присваивает уникальный числовой код. По сути, она превращает ваши слова в последовательность чисел, с которыми и производит все дальнейшие вычисления.

Осознание этого факта помогает понять две важные вещи. Во-первых, именно поэтому все технические ограничения, например, максимальная длина запроса, измеряются в токенах – модель оперирует именно этими числовыми единицами. А во-вторых, свой ответ она строит, анализируя последовательность этих чисел и вычисляя, какое число с наибольшей вероятностью должно пойти следующим. Затем это

число-прогноз преобразуется обратно в знакомый нам токен — слово или его часть.

Важно понимать, что этот процесс генерации не является **жестко предопределенным**. Это не математическая формула, у которой есть только один верный ответ. Если бы модель всегда выбирала только один, самый вероятный следующий токен, ее ответы были бы сухими, однообразными и совершенно лишенными гибкости.

Чтобы управлять этим, существуют параметры, которые вы, как специалист, можете контролировать. Основной из них — **"температура"**. Представьте ее как своего рода "регулятор креативности". Если вы установите низкое значение температуры, скажем, 0.2, модель будет вести себя очень консервативно и почти всегда выбирать самый очевидный, самый вероятный следующий токен. Такой режим идеален для задач, где нужна максимальная точность: извлечение фактов из текста, написание программного кода, составление кратких резюме. Если же вы повысите температуру до 0.8 или выше, вы позволите модели рассматривать и менее очевидные варианты. Вы как бы даете ей разрешение на риск. Это делает ее ответы более творческими, разнообразными и неожиданными, что отлично подходит для мозгового штурма, написания художественных текстов или поиска нестандартных идей.

Именно потому, что весь процесс сводится к управляемой генерации вероятностных последовательностей токенов,

качество вашего первоначального запроса имеет решающее значение. Этот запрос, который мы называем **промт**, – это не просто вопрос. Это полный набор начальных условий, которые вы задаете для всей последующей работы модели. И подход к созданию этого набора условий – это то, что отличает любителя от профессионала. Этот подход и называется **промт-инжиниринг**.

Слово "инжиниринг" здесь ключевое. Оно подразумевает системное проектирование, а не спонтанное творчество. Давайте рассмотрим это на конкретном, практическом примере.

Представим, что ваша задача – получить рекламный текст для нового протеинового батончика под названием "Импульс Заряд". Его характеристики: 25 грамм белка, низкое содержание сахара, вкус – "шоколадный фадж". Бренд "Импульс" общается с аудиторией в дерзком, энергичном и мотивирующем стиле.

Обычный пользователь написал бы что-то вроде: "напиши рекламный текст для протеинового батончика 'Импульс Заряд'". В ответ он получил бы безликий шаблонный текст, бесполезный для реального бизнеса.

Профессионал же подходит к задаче как к инженерному проекту. Его работа начинается с гипотезы: если я не просто дам задачу, а создам для модели полную рабочую среду, включающую роль, контекст и, главное, примеры нужного стиля, то результат будет на порядок качественнее. Осно-

вываясь на этом, он приступает к проектированию, то есть к конструированию сложного, многокомпонентного промта, где все части работают вместе.

Он не пишет отдельные команды, а собирает единую, логичную инструкцию. Этот процесс начинается с назначения **роли**, чтобы настроить модель на нужный тон: "Ты – дерзкий и мотивирующий копирайтер спортивного бренда 'Импульс'". Сразу за этим он предоставляет **контекст**, чтобы модель оперировала нужными фактами: "Мы запускаем новый продукт – протеиновый батончик 'Импульс Заряд' с 25 граммами белка, минимальным содержанием сахара и вкусом 'шоколадный фадж'".

Далее следует решающий шаг – предоставление **примеров**, чтобы модель могла скопировать нужный стиль, а не изобретать свой. Инженер добавляет: "Для калибровки стиля, вот три реальных примера наших текстов. Первый: 'Твоя тренировка окончена. Твои мышцы требуют топлива. Наш протеин 'Импульс Фьюжн' – это то, что им нужно. Не жди. Действуй.' Второй: 'Готов рвать железо? Предтренировочный комплекс 'Импульс Драйв' зажжет в тебе огонь. Максимальная отдача. Без компромиссов.' Третий: 'Хватит мечтать о результате. Начни его создавать. 'Импульс' – твоя энергия. Твоя сила. Твоя победа.'".

Только после такой подготовки дается прямая **инструкция**: "Теперь, основываясь на этом стиле, напиши рекламное описание для батончика 'Импульс Заряд'. Обязательно

используй короткие, рубленые предложения и обращайся к клиенту на 'ты'. И в конце устанавливаются четкие **ограничения** по формату: "Итоговый текст должен быть объемом не более 500 символов".

В результате этого процесса проектирования вместо простого вопроса у вас получается полноценное техническое задание. Вот как оно выглядит целиком:

Ты – дерзкий и мотивирующий копирайтер спортивного бренда 'Импульс'. Мы запускаем новый продукт – протеиновый батончик 'Импульс Заряд' с 25 граммами белка, минимальным содержанием сахара и вкусом 'шоколадный фадж'. Для калибровки стиля, вот три реальных примера наших текстов. Первый: 'Твоя тренировка окончена. Твои мышцы требуют топлива. Наш протеин 'Импульс Фьюжн' – это то, что им нужно. Не жди. Действуй.' Второй: 'Готов рвать железо? Предтренировочный комплекс 'Импульс Драйв' зажжет в тебе огонь. Максимальная отдача. Без компромиссов.' Третий: 'Хватит мечтать о результате. Начни его создавать. 'Импульс' – твоя энергия. Твоя сила. Твоя победа.'. Теперь, основываясь на этих примерах и стиле, напиши рекламное описание для нового батончика 'Импульс Заряд'. Обязательно используй короткие, рубленые предложения и обращайся к клиенту на 'ты'. Итоговый текст должен быть объемом не более 500 символов.

Получив результат, инженер анализирует его, и при необходимости дорабатывает промт, выдвигая новую гипотезу.

Этот системный, итеративный подход и есть суть инжиниринга.

Изначально существовало только взаимодействие через **жесткие команды**. Вспомните интерфейсы командной строки. Пользователь был обязан знать точный, формализованный язык машины и ее строгий синтаксис. Любая ошибка, любая опечатка приводила к отказу системы. В этой парадигме человек полностью подстраивался под машину. Свобода действия была равна нулю; вы могли выполнить только ту операцию, для которой существовала конкретная, заранее запрограммированная команда.

Затем наступила эра **предопределенного выбора**, которую принесли графические пользовательские интерфейсы. Иконки, кнопки и меню сделали компьютеры доступными для миллионов. Взаимодействие стало интуитивным, но его суть не изменилась кардинально. Свобода пользователя по-прежнему была ограничена, на этот раз – набором опций, которые предусмотрели для него разработчики. Вы не могли попросить программу сделать что-то новое; вы могли лишь выбрать один из предложенных вариантов.

Следующим шагом стало взаимодействие через **поиск по ключевым словам**, расцвет которого пришелся на эру поисковых систем. Человечество впервые начало массово использовать естественный язык для общения с машиной. Однако функция этой машины была ограничена поиском и извлечением уже существующей информации. Система не со-

здавала ничего нового. Она лишь сопоставляла слова в вашем запросе с документами в своей базе. Она не выполняла инструкций, а лишь указывала, где можно найти релевантную информацию.

И вот сегодня мы находимся в центре нового, четвертого этапа – этапа **взаимодействия на основе инструкций**. Современные генеративные модели впервые в истории сняли прежние ограничения. Вам больше не нужно знать формальный язык команд. Вас больше не сдерживает ограниченный набор кнопок. И вы больше не ограничены поиском существующего. Вы получили возможность формулировать уникальные, сложные, многоступенчатые задачи на своем естественном языке, и система будет пытаться их выполнить, генерируя абсолютно новый, ранее не существовавший контент. Произошел фундаментальный сдвиг: от выбора из предложенного – к созданию желаемого по вашему личному проекту.

Овладение этим новым типом взаимодействия – это не просто техническое упражнение. Оно преследует четыре четкие, прагматичные цели, которые и определяют ценность промт-инженера как специалиста.

Первая цель – **качество**! Ваша задача – научиться получать от нейросети результат, который не просто "годится", а соответствует высоким профессиональным стандартам. Это значит, что текст будет стилистически выверенным, точным и убедительным. Код – чистым, эффективным и хорошо про-

комментированным. А бизнес-стратегия – детальной и логичной. Вы стремитесь сделать так, чтобы результат работы ИИ был неотличим от продукта высококлассного специалиста.

Вторая цель – **надежность**! Профессиональная работа не терпит случайности. Цель промт инжиниринга – превратить процесс получения результата в предсказуемую процедуру. Вы должны быть уверены, что разработанный вами промт сегодня, завтра и через месяц будет давать стабильно качественный результат. Именно надежность позволяет встраивать генеративные модели в реальные бизнес-процессы.

Третья цель – **эффективность**! Это прямой экономический показатель вашей работы. Неэффективный подход – это часы, потраченные на метод проб и ошибок, десятки неудачных попыток и постоянное разочарование. Эффективный инженерный подход – это получение нужного результата с первой или второй попытки. Это экономия вашего самого ценного ресурса – времени. В масштабах компании это означает ускорение проектов, повышение производительности и, как следствие, рост прибыли.

Четвертая, высшая цель – это **инновации**! Это способность не просто решать известные задачи быстрее, а находить для нейросети совершенно новые применения. Профессионал не боится экспериментировать. Он комбинирует разные техники, чтобы заставить модель решать комплексные проблемы, о которых ее создатели, возможно, и не ду-

мали. Именно так рождаются новые рабочие методики, создаются уникальные продукты и достигается реальное конкурентное преимущество.

Итак, мы заложили фундамент. Мы определили, что нейросеть – это управляемая вероятностная система. Мы увидели, что промт-инжиниринг – это системный подход к проектированию инструкций для этой системы, который позволяет добиваться высокого качества, надежности, эффективности и открывает путь к инновациям.

В следующей главе мы начнем детальный разбор анатомии промта и тех конкретных, воспроизводимых техник, которые станут основой вашей повседневной работы.

Глава 2: Анатомия идеального промта. Ключевые компоненты.

В предыдущей главе мы определили, *что* такое промт-инжиниринг и *почему* это ключевой навык для эффективной работы с нейросетями. Теперь мы переходим от теории к практике. Пришло время разобраться, *как* именно конструировать те самые мощные и точные инструкции, которые дадут предсказуемый результат.

Секрет профессионального подхода не в поиске неких волшебных слов, а в понимании структуры. Любой эффективный промт – это не просто случайный набор предложений, а четко спроектированный механизм. В этой главе мы препарируем этот механизм и изучим его анатомию. Мы разберем шесть ключевых компонентов, из которых, как из

деталей конструктора, собирается практически любая безупречная инструкция для нейросети.

Освоив эти шесть элементов, вы перестанете действовать наугад. У вас появится системный подход, своего рода чек-лист, который позволит вам уверенно и быстро создавать промты для любых, даже самых сложных задач. Каждый из этих компонентов – это отдельный рычаг управления, позволяющий контролировать тон, содержание, стиль и структуру ответа нейросети.

Итак, давайте начнем с первого и, возможно, самого недооцененного компонента, который задает тон всему дальнейшему взаимодействию и мгновенно повышает качество результата.

Компонент первый! Роль.

Первый и часто самый недооцененный компонент – это **Роль**. Задавая нейросети роль, вы, по сути, даете ей "профессию" и "личность". Вы перестаете обращаться к безликой программе и начинаете говорить с экспертом в нужной вам области. Это невероятно мощный прием, который мгновенно повышает качество и релевантность ответа. Почему? Потому что модель, получив роль, активирует те кластеры данных и те языковые шаблоны, которые наиболее соответствуют этой роли. Она начинает "думать" и говорить как тот, кем вы ее назначили.

Давайте посмотрим на наглядном контрасте.

Предположим, вам нужно понять, что такое технология

блокчейн.

Плохой промт (без роли):

Объясни, что такое блокчейн.

В ответ вы получите сухой, академический текст, скорее всего, скопированный из условной Википедии. Он будет правильным, но сложным для понимания и лишенным практической ценности для конкретного человека.

Теперь давайте применим наш первый компонент и зададим модели роль.

Хороший промт (с ролью):

Ты – опытный финансовый консультант. Твой клиент – консервативный инвестор в возрасте 50 лет, который хочет вложить деньги во что-то надежное, но совершенно не разбирается в технологиях. Объясни ему простыми словами, используя аналогии из реального мира, что такое блокчейн и почему эта технология считается безопасной для хранения данных.

Почувствуйте разницу. Результат изменится кардинально. Нейросеть откажется от сложной терминологии. Она будет использовать аналогии, понятные инвестору. Она сделает акцент не на технических деталях, а на том, что волнует клиента – на **безопасности и надежности**. Вы получите не просто информацию, а инструмент для решения вашей конкретной задачи – убедить клиента.

Назначая роль, вы можете быть кем угодно: маркетологом, ученым-физиком, сценаристом, детским психологом, опыт-

ным шеф-поваром. Чем точнее и детальнее вы опишете роль, тем глубже модель в нее погрузится.

Компонент второй! Контекст.

Контекст – это вся фоновая информация, все вводные данные и обстоятельства, которые необходимы модели для понимания вашей ситуации. Нейросеть не обладает телепатией. Она не знает, над какой задачей вы работаете и что происходило до того, как вы написали промт. Предоставление контекста – это ваша обязанность. Чем больше релевантных деталей вы дадите, тем меньше модели придется додумывать, и тем точнее будет ее ответ.

Снова посмотрим на пример.

Задача: уведомить клиента о задержке поставки.

Плохой промт (без контекста):

Напиши письмо клиенту о том, что мы задерживаем поставку.

Это ужасный промт. Модель не знает ничего: кто вы, кто клиент, что за поставка, на сколько задержка, какова причина. В лучшем случае она сгенерирует предельно общий и бесполезный шаблон, который вызовет у клиента только раздражение.

Теперь добавим исчерпывающий контекст.

Хороший промт (с контекстом):

Мы – производственная компания "ТехноСталь". Наш клиент – строительная компания "Мегаполис", с которой мы работаем уже пять лет. 10 октября мы должны были

отгрузить им партию арматуры (заказ №78-22) для строительства важного объекта. Из-за сбоя на производственной линии мы не успеваем в срок. Новая дата отгрузки – 18 октября. Для нас критически важно сохранить хорошие отношения с этим клиентом.

Видите? Теперь у модели есть все необходимые данные. Она знает участников, историю отношений, детали заказа, причину и длительность задержки, а главное – вашу цель (сохранить хорошие отношения). Таким образом вы устраняете неопределенность и даете модели четкие рабочие параметры, необходимые для подготовки точной и выверенной коммуникации.

Компонент третий! Задача.

Теперь, когда у нас есть роль и контекст, нам нужна – **Задача**. Это самый прямой и очевидный компонент промта. Здесь вы четко и недвусмысленно указываете, что именно должна сделать нейросеть. Ошибка многих заключается в том, что они формулируют задачу слишком расплывчато. "Напиши об этом", "проанализируй это", "предложи что-нибудь" – это не задачи, а приглашение к свободной беседе. Инженерный подход требует максимальной конкретики.

Давайте объединим наш предыдущий пример с контекстом и посмотрим, как четкая постановка задачи меняет дело.

Плохой промт (с контекстом, но с нечеткой задачей):

Мы – производственная компания "ТехноСталь". Наш клиент – строительная компания "Мегаполис", с которой мы работаем уже пять лет. 10 октября мы должны были отгрузить им партию арматуры (заказ №78-22) для строительства важного объекта. Из-за сбоя на производственной линии мы не успеваем в срок. Новая дата отгрузки – 18 октября. Для нас критически важно сохранить хорошие отношения с этим клиентом. Напиши им об этом.

"Напиши им об этом" – это очень плохая инструкция. Что именно написать? В каком тоне? Что предложить?

А вот как выглядит промт с четко сформулированной задачей.

Хороший промт (с контекстом и четкой задачей):

Мы – производственная компания "ТехноСталь". Наш клиент – строительная компания "Мегаполис", с которой мы работаем уже пять лет. 10 октября мы должны были отгрузить им партию арматуры (заказ №78-22) для строительства важного объекта. Из-за сбоя на производственной линии мы не успеваем в срок. Новая дата отгрузки – 18 октября. Для нас критически важно сохранить хорошие отношения с этим клиентом. **Твоя задача: составить проект официального письма на имя руководителя отдела снабжения "Мегаполиса", Виктора Арсеньева. В письме нужно:**

Первое! Принести извинения за срыв сроков.

Второе! Четко указать причину задержки и новую дату поставки.

Третье! Подчеркнуть, что мы ценим наше сотрудничество.

Четвёртое! В качестве компенсации предложить бесплатную доставку этой партии.

Теперь у модели нет пространства для неверных интерпретаций. У нее есть пошаговый план действий. Она точно знает, что, кому и в какой последовательности нужно написать. Результат будет на 99% соответствовать вашим ожиданиям с первой же попытки.

Компонент четвертый! Примеры.

Это, возможно, самый мощный компонент во всем вашем арсенале. Если Роль, Контекст и Задача *объясняют* модели, что вы хотите, то **Примеры** *показывают* ей это. Этот метод, известный как "few-shot prompting" (промтинг с несколькими примерами), часто работает эффективнее любых, даже самых детальных инструкций. Предоставляя модели 1-3 примера желаемого результата, вы даете ей образец для подражания. Она анализирует структуру, стиль, тон ваших примеров и генерирует ответ по аналогии.

Этот компонент незаменим, когда вам нужно добиться очень специфического или креативного стиля.

Представим, что вы ведете интернет-магазин необычных

подарков и вам нужны описания для товаров в особом, интригующем стиле.

Плохой промт (без примеров):

Напиши креативное и загадочное описание для товара "Планер с черными страницами и белой ручкой".

Результатом станет поверхностный и предсказуемый текст. Он будет состоять из общих фраз о "раскрытии творческого потенциала", "белых чернилах на черном фоне" или "сохранении ваших самых сокровенных мыслей". Это будет грамотный, но совершенно "пустой" текст, лишенный уникального стиля, который мог бы выделить товар на фоне других.

Теперь давайте покажем модели, чего именно мы хотим.

Хороший промт (с примерами):

Ты – философ-маркетолог для магазина необычных подарков. Твоя цель – не просто описывать товары, а раскрывать их скрытый смысл в короткой, интригующей форме. Для калибровки стиля, вот два примера.

Пример первый, для товара "Чашка с картой созвездий": 'Днем – это просто черная чашка. Но налейте в нее кипяток, и она раскроет свой секрет – на стенках проступит точная карта звездного неба. Не просто посуда, а ваш личный фрагмент космоса, который просыпается от тепла.'

Пример второй, для товара "Магнитный песок": Он не подчиняется гравитации. Он тянется к магниту, образуя инопланетные пейзажи. Глупая игрушка? Нет. Это физика,

с которой можно играть.

Теперь, используя в точности этот же стиль, напиши описание для нового товара: "Планер с черными страницами и белой ручкой".

Ответ, сгенерированный по такому промту, будет совершенно иным. Модель уловит структуру: короткие предложения, контраст, легкая ирония. Она выдаст что-то вроде: "Обычные планеры ждут черных чернил. Этот – ждет белых. Не просто блокнот, а возможность начать свою историю с чистого, но абсолютно черного листа. Страшно? Нет. Необычно."

Вы получаете желаемый стиль не потому, что вы его описали, а потому, что вы его показали.

Компонент пятый! Формат.

Этот компонент отвечает за структуру вашего ответа. Если вы не укажете формат, нейросеть по умолчанию выдаст вам сплошной абзац текста. Но в большинстве рабочих задач нам нужна информация в строго определенном, структурированном виде. **Формат** – это ваша инструкция по "упаковке" результата.

Предположим, вам нужно быстро извлечь суть из отзыва клиента.

Плохой промт (без указания формата):

Проанализируй этот отзыв и выдели основное: "Отель в целом понравился. Расположение шикарное, прямо в центре. Номер был чистый, свежий ремонт. Но завтрак не очень

– выбора почти нет, а кофе был холодный. Еще и Wi-Fi в номере иногда отключался. Но персонал очень быстро решал все проблемы. Это плюс."

Вы получите краткий пересказ отзыва в виде одного-двух предложений. Эту информацию все еще придется обрабатывать вручную.

Теперь давайте потребуем от модели нужный нам формат.

Хороший промт (с четким форматом):

Проанализируй этот отзыв и выдели основное: "Отель в целом понравился. Расположение шикарное, прямо в центре. Номер был чистый, свежий ремонт. Но завтрак не очень – выбора почти нет, а кофе был холодный. Еще и Wi-Fi в номере иногда отключался. Но персонал очень быстро решал все проблемы. Это плюс." **Твоя задача: предоставить ответ в виде таблицы Markdown с двумя колонками: "Плюсы" и "Минусы". Каждая колонка должна содержать пункты, извлеченные из текста.**

Результат будет идеально структурирован. Вы получите готовую таблицу, которую можно сразу скопировать в отчет или отправить коллеге. Точно так же вы можете запрашивать формат нумерованного списка, HTML-кода, JSON-объекта, XML-разметки – чего угодно. Четкое указание формата экономит массу времени на последующей обработке информации.

Компонент шестой! Ограничения.

Последний, но не по значению, компонент – это **Огра-**

ничения. Если задача говорит модели, что *нужно* сделать, то ограничения говорят, чего делать *не нужно*, или в какие рамки себя поставить. Это ваши "стоп-сигналы" и "ограждения", которые делают результат еще более точным и соответствующим вашим требованиям. Ограничения могут касаться объема, стиля, содержания и многого другого.

Задача: придумать идеи для детского праздника.

Плохой промт (без ограничений):

Придумай 5 идей для дня рождения девочки 9 лет.

Вы рискуете получить совершенно нерелевантные предложения: от полета на воздушном шаре (слишком дорого) до квеста по Гарри Поттеру (которого ребенок его не любит).

Теперь добавим конкретные рамки.

Хороший промт (с ограничениями):

Придумай 5 идей для празднования дня рождения девочки, которой исполняется 9 лет. **Важные ограничения:** бюджет мероприятия – до трёхсот долларов. Праздник будет проходить дома, в двухкомнатной квартире, поэтому идеи не должны требовать много места. Именинница не любит принцесс и фей, поэтому избегай этих тем. Предложи варианты, которые включают творческий или научный элемент.

Такой промт отсекает неподходящие варианты. Модель будет работать в узком коридоре ваших требований и сгенерирует практичные, релевантные и персонализированные идеи, например, "мастер-класс по созданию слаймов" или

"домашний научный квест с простыми химическими экспериментами".

Собираем все вместе!

Теперь, когда мы знаем все шесть компонентов, давайте посмотрим на разительный контраст между обычным запросом и полноценным инженерным промтом, в котором работают все элементы.

Задача: Руководитель просит вас сделать краткую выжимку из длинного еженедельного отчета о продажах.

ПЛОХОЙ ПРОМТ:

Сделай краткую сводку по этому отчету:

ИДЕАЛЬНЫЙ ПРОМТ:

[Роль] Ты – ведущий бизнес-аналитик, который готовит сводку для генерального директора. Твой стиль – предельно четкий, лаконичный и сфокусированный на ключевых показателях. **[Контекст]** Вот текст еженедельного отчета по продажам за сентябрь 2025 года. Сюда вставляете сам отчет. **[Пример]** Для справки, вот как выглядела сводка за прошлый месяц: Сюда вставляете прошлый отчет" **[Задача]** Твоя задача – проанализировать предоставленный отчет и выделить 2-3 главных достижения и 1-2 самые острые проблемы за неделю. **[Формат]** Представь ответ в виде маркированного списка под двумя заголовками: "Главные достижения:" и "Проблемные зоны:". **[Ограничения]** Вся сводка должна быть не длиннее 100 слов. Используй только деловой тон, без эмоциональных оценок и сложных терми-

нов.

Разница в результате будет колоссальной. Вместо размытого пересказа вы получите структурированный, профессиональный документ, готовый для отправки руководству.

Вы познакомились с шестью деталями конструктора. Роль, Контекст, Задача, Примеры, Формат и Ограничения. Теперь вы знаете, как из них собрать мощный инструмент.

Глава 3: Стил, тон и аудитория. Как научить нейросеть говорить на вашем языке?

В предыдущей главе мы досконально разобрали шесть структурных компонентов, из которых собирается эффективный промт. Освоение этих компонентов позволяет получать технически правильный, точный и релевантный результат, соответствующий поставленной задаче.

Однако в профессиональной коммуникации этого часто недостаточно. Текст может быть формально верным, но при этом неэффективным.

Теперь мы переходим на следующий уровень – от управления содержанием к управлению подачей. В этой главе мы сосредоточимся на том, как управлять стилем, тоном и адаптацией текста для конкретной аудитории. Если компоненты из прошлой главы отвечали за то, **ЧТО** говорит нейросеть, то техники из этой главы отвечают за то, **КАК** она это говорит. От этого "как" напрямую зависит, будет ли ваш текст убедительным, вызовет ли он доверие и достигнет ли он своей конечной цели.

Эта глава посвящена тому, как научить нейросеть говорить не просто правильно, а именно так, как нужно вам – для вашей цели и, что самое главное, для вашей аудитории.

Мы начнем с самого главного – с управления эмоциональной окраской текста.

Задаём тональность! Эмоциональный окрас вашего текста.

Тональность или **тон** – это настроение и атмосфера вашего текста. Одна и та же информация, поданная в разном тоне, может произвести на читателя совершенно противоположное впечатление. Умение управлять тональностью – это ключ к созданию текстов, которые не просто информируют, а убеждают, увлекают и вызывают нужные вам эмоции.

Нейросеть по умолчанию стремится к нейтральному, слегка формальному стилю. Ваша задача как инженера – задать именно ту тональность, которая требуется для вашей задачи. Делается это с помощью прямых указаний и прилагательных, описывающих нужный стиль.

Давайте рассмотрим, как меняется результат в зависимости от тональности, на одном и том же примере.

Исходная задача: Написать текст о новой функции в приложении для управления проектами. Функция называется "AI Ассистент" и она автоматически создает краткие сводки из больших задач.

Пример первый! Официальный тон.

Представим, что нам нужно составить объявление для

рассылки крупным корпоративным клиентам. Здесь важны сдержанность, профессионализм и акцент на бизнес-выгоде.

Плохой промт (без указания тона):

Напиши текст о нашей новой функции "AI Ассистент".

Результат будет слишком простым и недостаточно убедительным для серьезной аудитории.

Хороший промт (с официальным тоном):

Ты – директор по продукту в IT-компании. Твоя задача – составить официальное объявление для наших корпоративных клиентов о запуске новой функции "AI Ассистент". Тон должен быть строго деловым, профессиональным и лаконичным. Сделай акцент на повышении продуктивности команд и безопасности обработки данных.

Вероятный результат: "Уважаемые партнеры, информируем вас о внедрении нового модуля 'AI Ассистент' в нашу платформу. Данное решение предназначено для автоматического формирования сводок по задачам, что позволяет оптимизировать рабочие процессы и сократить время на менеджмент проектов. Все данные обрабатываются в рамках защищенного контура вашей компании."

Такой текст идеально подходит для официальной коммуникации.

Пример второй! Дружелюбный и энергичный тон.

Теперь представим, что мы пишем пост об этой же функции, но для своего телеграм-канала, где мы общаемся с пользователями неформально.

Плохой промт:

Напиши пост для телеграм-канала о новой функции "AI Ассистент".

Текст получится сухим и не вызовет интереса у подписчиков, привыкших к легкому стилю общения.

Хороший промт (с дружелюбным тоном):

Ты – SMM-менеджер, который влюблен в свой продукт. Напиши короткий и энергичный пост для нашего телеграм-канала о выходе новой потрясающей функции "AI Ассистент". Тон – максимально дружелюбный, восторженный и простой. Используй короткие, яркие фразы, обращайся к подписчикам на "ты" и объясни, как эта штука сделает их жизнь проще.

Вероятный результат: «Друзья, у нас потрясающая новость! Устали читать бесконечные описания задач? Мы тоже! Поэтому мы создали 'AI Ассистента'. Теперь наш искусственный интеллект сам прочитает всю рутину и за секунду выдаст вам краткую суть. Больше времени на творчество, меньше – на скучные отчеты! Попробуйте прямо сейчас!»

Этот текст заряжен энергией и отлично сработает в соцсетях, даже без использования эмодзи.

Пример третий! Экспертный и убедительный тон.

А теперь напишем статью об этой же функции для профильного IT-блога. Наша цель – не просто объявить, а убедить технически подкованную аудиторию в превосходстве нашего решения.

Плохой промт:

Напиши статью в блог о функции "AI Ассистент".

Статья будет поверхностной, без глубоких аргументов.

Хороший промт (с экспертным тоном):

Ты – руководитель продукта, который досконально знает все технические и пользовательские преимущества новой функции "AI Ассистент". Напиши статью в наш технический блог для IT-специалистов и менеджеров проектов. Тон должен быть экспертным, уверенным и сфокусированным на ценности для пользователя. Объясни, почему наш подход к автоматической суммаризации эффективнее решений конкурентов, сделав акцент на нашей уникальной архитектуре обработки контекста. Приведи данные, как функция сокращает управленческую нагрузку на 15% в командах от 10 человек.

Вероятный результат: "Современные подходы к автоматической суммаризации часто сталкиваются с проблемой потери контекста в длинных задачах. Наш новый 'AI Ассистент' решает эту проблему за счет использования гибридной архитектуры, совмещающей трансформер и графовые сети. Это позволяет модели сначала выстраивать карту взаимосвязей внутри проекта, и только затем проводить детальную суммаризацию. Тесты показывают, что такой подход не только повышает точность сводки, но и сокращает среднее время на менеджмент задач до 15%".

Этот текст вызывает доверие и демонстрирует глубокую

экспертизу.

Как видите, прямое указание на желаемую тональность – это простой, но невероятно эффективный способ управлять восприятием вашего текста.

Теперь, когда мы научились задавать настроение, пора сделать следующий шаг – научиться говорить на одном языке с конкретными группами людей.

Определение целевой аудитории!

Задать правильный тон – это половина успеха. Вторая половина – убедиться, что этот тон, а также сами слова, примеры и аргументы идеально подходят тем людям, для которых вы пишете. Это подводит нас к следующему ключевому понятию – **целевой аудитории**.

Вы никогда не будете говорить одинаково с ребенком, с начальником на совещании и с лучшим другом в баре. Вы интуитивно подбираете слова, сложность фраз и стиль общения. Точно так же должен поступать и промт-инженер. Указывая нейросети, для кого предназначен текст, вы позволяете ей произвести тонкую настройку. Модель начинает подбирать лексику, примеры и аргументы, которые найдут наибольший отклик именно у этой группы людей.

Описание аудитории в промте может включать:

Возраст и поколение.

Профессию и уровень знаний в теме.

Цели и мотивацию.

Проблемы и "боли".

Давайте посмотрим на одном примере, как текст на одну и ту же тему кардинально меняется при смене целевой аудитории.

Исходная задача: Написать статью, объясняющую преимущества интервального голодания.

Пример первый! Аудитория – молодые фитнес-энтузиасты.

Представим, что мы пишем пост для популярного спортивного паблика. Аудитория здесь – молодые люди, которые следят за трендами и хотят получить максимальный результат от тренировок.

Плохой промт (без указания аудитории):

Напиши статью о пользе интервального голодания.

Статья получится слишком общей, научной и скучной для этой аудитории. Она не будет использовать их язык и не затронет их цели.

Хороший промт (с описанием аудитории):

Ты – популярный фитнес-блогер. Твоя аудитория – парни и девушки от 20 до 30 лет, которые регулярно тренируются в зале и интересуются способами улучшить свою форму и результаты. Напиши для них пост в блог об интервальном голодании. Используй простой, энергичный и немного сленговый язык (например, слова 'сушка', 'метаболизм', 'рельеф'). Объясни, как этот метод помогает эффективно сжигать жир, не теряя при этом мышечную массу, и как он может повысить энергию на тренировках.

Пример второй! Аудитория – люди старшего возраста, заботящиеся о здоровье.

Теперь представим, что мы пишем статью для сайта о здоровом образе жизни. Наша аудитория здесь – люди более зрелые, которых интересует не рельеф, а долголетие, профилактика заболеваний и научный подход.

Плохой промт:

Напиши статью о пользе интервального голодания.

Тот же плохой промт даст здесь еще худший результат, так как молодежный сленг или фокус на мышцах будут абсолютно неуместны.

Хороший промт (с другой аудиторией):

Ты – врач-диетолог. Твоя аудитория – мужчины и женщины в возрасте от 45 до 60 лет, которые ищут проверенные и безопасные способы улучшить свое здоровье и самочувствие. Напиши для них статью об интервальном голодании. Объясняй сложные вещи простыми словами, но опирайся на научные понятия (например, аутофагия, инсулин). Сделай акцент не на быстром похудении, а на долгосрочных преимуществах: улучшении обмена веществ, снижении уровня сахара в крови и клеточном омоложении.

Заключение!

Тон и Аудитория – это два неразрывно связанных компонента. Вы не можете выбрать правильный тон, не зная, для кого вы пишете. И вы не сможете достучаться до аудитории, если будете говорить с ней в неподходящем тоне.

Всегда начинайте с вопроса: "Кто мой читатель и какого результата я хочу от него добиться?". Ответ на этот вопрос подскажет вам, какой тон выбрать и как описать аудиторию в вашем промте.

Освоив эти два аспекта, вы, по сути, превращаете нейросеть в целую команду узкоспециализированных авторов. У вас появляется копирайтер для молодежи, научный журналист для экспертов, заботливый врач для пациентов, строгий юрист для деловых партнеров. Вы получаете полный контроль над стилем и восприятием генерируемого контента.

Глава 4: Сила контекста и последовательности. Ведение диалога с нейросетью.

До этого момента мы рассматривали промт как одноразовый, завершённый запрос. Вы конструировали его, отправляли и получали результат. Но настоящая технология – раскрывается, когда вы начинаете воспринимать чат с нейросетью не как серию отдельных вопросов, а как единую рабочую сессию, один непрерывный разговор.

Каждый успешный диалог с ИИ строится на фундаментальном понятии – **контексте**. Понимание того, как нейросеть работает с контекстом, мгновенно переведет вас на новый уровень мастерства.

Как нейросеть запоминает историю диалога?

Вы наверняка замечали, что в рамках одного разговора нейросеть помнит ваши предыдущие реплики. Вы можете сказать: "Придумай идею для бизнеса", а в следующем сооб-

щении написать "Отлично, а теперь составь для этой идеи бизнес-план", и она поймет, о какой именно идее идет речь.

Как это работает? Обладает ли она памятью? И да, и нет. У модели нет долгосрочной памяти в человеческом понимании. Вместо этого у нее есть "**окно контекста**" – своего рода оперативная, кратковременная память.

Представьте это так: перед тем, как ответить на ваше новое сообщение, нейросеть заново с огромной скоростью перечитывает всю вашу недавнюю переписку в текущем чате. Ваш новый промт и несколько предыдущих сообщений (ваших и ее) объединяются в один большой мета-промт. Именно на основе этого полного текста она и генерирует следующий ответ. Благодаря этому и создается ощущение непрерывного, осмысленного диалога.

Однако ключевой момент, который нужно осознать – **эта память не бесконечна**. Окно контекста имеет строго ограниченный размер, который измеряется в токенах. У разных моделей он разный – от нескольких тысяч до сотен тысяч токенов. Что это значит на практике?

Это значит, что если ваш диалог станет слишком длинным, самые первые сообщения начнут "выпадать" из этого окна. Нейросеть их попросту перестанет "видеть", когда будет генерировать новый ответ. Она их "забудет".

Давайте рассмотрим на простом примере.

Начало диалога:

Вы: Давай создадим концепцию мобильного приложения.

Оно должно помогать людям изучать редкие исторические факты. Назовем его "Хронограф". Целевая аудитория – студенты и любители истории. Ключевая особенность – ежедневные короткие статьи в формате сторис.

Продолжение (через 2-3 сообщения):

Вы: Отлично. Теперь предложи три варианта монетизации для "Хронографа".

Модель без проблем справится с этой задачей. Вся информация о названии, аудитории и ключевой особенности все еще находится в ее окне контекста. Она "помнит" все детали проекта.

Гипотетическое продолжение (через 50 сообщений): А теперь представьте, что вы продолжили диалог. Вы долго обсуждали дизайн, стек технологий, маркетинговую стратегию, нанимали виртуальных сотрудников, написали десятки страниц текста... Окно контекста заполнилось этой новой информацией, и ваше самое первое сообщение о концепции "Хронографа" было вытеснено за его пределы.

И вот, в конце этого длинного разговора, вы пишете:

Вы: Хорошо, а теперь напиши короткий рекламный слоган для нашего приложения.

С высокой долей вероятности модель может выдать что-то общее или даже переспросить: "Конечно! Уточните, пожалуйста, о каком приложении идет речь и каковы его ключевые особенности?". Она "забыла" самые первые, базовые инструкции, потому что они больше не попадают в ее поле

зрения.

Понимание этого механизма – первый шаг к управлению диалогом. Вы начинаете осознавать, что контекст – это ценный и ограниченный ресурс. В следующей части мы поговорим о том, как активно управлять этим ресурсом: как уточнять и улучшать ответы, не теряя контекста, и как "напоминать" модели о ключевых инструкциях, даже если диалог стал очень длинным.

Искусство итерации!

Самая большая ошибка, которую можно совершить в работе с нейросетью – это принять ее первый ответ за финальный. Профессионал никогда так не делает. Он относится к первому сгенерированному тексту как к сырью, как к куску глины, которому еще только предстоит придать идеальную форму. Процесс доработки ответа через серию уточняющих команд называется **итеративным улучшением**.

Это и есть настоящий диалог. Вы не просто даете команду, вы становитесь режиссером, который работает с актером. Вы говорите: "Хорошо, а теперь давай тот же монолог, но с другим настроением", "Здесь добавь паузу", "А этот фрагмент нужно полностью переделать".

Вот несколько базовых, но невероятно мощных команд, которые должны стать частью вашего инструментария:

"Перепиши это, но..." – ваша основная команда для внесения правок. После "но" вы можете добавить любое условие:

"...сделай тон более деловым."

"...упрости этот абзац так, чтобы понял десятилетний ребенок."

"...добавь больше статистических данных."

"...замени все прилагательные на более яркие и эмоциональные."

"Развей эту мысль" или "Дополни" – используется, когда идея хороша, но изложена слишком кратко. Вы можете указать на конкретный абзац или предложение и попросить раскрыть его подробнее.

"Сократи" или "Перепиши без воды" – незаменимая команда, когда ответ получился слишком длинным и многословным. Вы можете попросить сократить текст до определенного количества символов или просто убрать все лишнее, оставив только суть.

Давайте посмотрим, как это работает на практике.

Задача: Написать приветственное письмо для новых подписчиков рассылки о личных финансах.

Шаг 1. Первый запрос:

Вы: Напиши приветственное письмо для новых подписчиков моей рассылки о личных финансах "Финансовый компас".

Шаг 2. Первый ответ ИИ:

"Здравствуйте! Благодарим вас за подписку на нашу рас-

сылку 'Финансовый компас'. В наших письмах вы найдете полезную информацию об управлении бюджетом, инвестициях и финансовых целях. Оставайтесь с нами! С уважением, команда 'Финансового компаса'."

Результат неплохой, но безликий и скучный.

Шаг 3. Первая итерация (меняем тон и добавляем детали):

Вы: Неплохо. А теперь **перепиши это, но** сделай тон гораздо более дружелюбным и личным, обращай на "ты". И еще **добавь** анонс того, что подписчик получит в первых трех письмах.

Шаг 4. Второй ответ ИИ:

"Привет! Я очень рад, что ты присоединился к 'Финансовому компасу'. Спасибо за доверие! Я верю, что управлять деньгами может каждый, и моя цель – помочь тебе в этом. В ближайших письмах мы начнем с основ: сначала разберем, как составить личный бюджет и перестать жить от зарплаты до зарплаты; затем поговорим о том, как создать финансовую подушку безопасности; а в третьем письме сделаем первые шаги в мир инвестиций. Готов навести порядок в своих финансах?"

Уже гораздо лучше! Текст стал личным и более содержательным. Но можно еще улучшить.

Шаг 5. Вторая итерация (финальные штрихи):

Вы: Отлично! Теперь **добавь** в конце призыв к действию: попроси подписчика ответить на это письмо, кратко расска-

зав о своей главной финансовой цели. Это поможет мне лучше адаптировать контент.

Шаг 6. Финальный результат:

"Привет! Я очень рад, что ты присоединился к 'Финансовому компасу'. Спасибо за доверие! Я верю, что управлять деньгами может каждый, и моя цель – помочь тебе в этом. В ближайших письмах мы начнем с основ: сначала разберем, как составить личный бюджет; затем поговорим о создании подушки безопасности; а после сделаем первые шаги в мир инвестиций.

P.S. Я буду очень благодарен, если ты ответишь на это письмо и в двух словах расскажешь о своей главной финансовой цели. Это поможет мне сделать рассылку еще полезнее именно для тебя!"

Вот это – идеальный результат. Мы получили его не с первого раза, а путем последовательных уточнений. Именно такой итеративный подход и отличает профессионала.

Активное управление контекстом!

Теперь вернемся к проблеме "забывания" в длинных диалогах. Как заставить модель помнить о самых важных инструкциях, даже если они были даны 50 сообщений назад? Ответ прост: нужно периодически "освежать" ключевую информацию в ее окне контекста.

Самый надежный способ – это **промт-суммаризация**.

Перед тем, как дать новую комплексную задачу в длинном чате, начните свой промт с короткой сводки-напоминания.

Вспомним наш пример с приложением "Хронограф".

Плохой промт (после долгого диалога):

Хорошо, а теперь напиши короткий рекламный слоган для нашего приложения.

Хороший промт (с напоминанием):

Продолжаем работу над проектом. Напоминаю ключевые детали: мы создаем мобильное приложение 'Хронограф' для изучения редких исторических фактов. Наша аудитория – студенты и эрудиты. Главная фишка – ежедневные факты в формате сторис. Тон общения – современный и увлекательный. Теперь, с учетом всего этого, придумай пять вариантов рекламных слоганов.

Такой промт практически гарантирует, что модель не потеряет нить и будет работать с той информацией, которая важна вам именно сейчас. Вы как бы берете самые важные файлы и кладете их на самый верх стола, чтобы они всегда были перед глазами.

Заключение!

Освоив техники ведения диалога, итеративного улучшения и управления контекстом, вы совершаете фундаментальный сдвиг. Нейросеть становится вашим полноценным рабочим партнером. Партнером, которого можно и нужно направлять, обучать и корректировать для достижения сложных, многоступенчатых целей.

Глава 5: Основные техники промтинга: Zero-Shot, One-Shot и Few-Shot Prompting.

В предыдущих главах мы научились конструировать промты из отдельных компонентов, задавать им нужный тон и вести последовательный диалог. Теперь мы сделаем шаг назад, чтобы увидеть картину целиком. Мы изучим три основополагающих способа постановки задачи для нейросети. По сути, это три главных "режима", в которых вы можете с ней общаться. Все, что мы изучали до этого, – это способы сделать каждый из этих режимов более точным и эффективным.

Понимание разницы между этими техниками и умение выбирать нужную в зависимости от ситуации – это то, что окончательно переведет вас из категории "продвинутых пользователей" в категорию настоящих промт-инженеров. Это даст вам то самое чувство полного контроля, когда вы заранее знаете, как подойти к задаче, чтобы получить предсказуемый результат.

Техника первая! Zero-Shot.

Zero-Shot Prompting (промтинг "с нулевым выстрелом" или без примеров) – это самый распространенный и интуитивно понятный способ взаимодействия с нейросетью. Вы используете его каждый раз, когда просто даете модели задачу, не предоставляя ей никаких образцов для подражания.

Суть техники: Вы полностью полагаетесь на те знания и умения, которые уже были заложены в модель во время ее масштабного обучения. Вы как бы говорите: "Ты – невероятно мощная языковая модель, тебя обучили на всей библио-

теке человечества. Я верю, что ты уже знаешь, как выполнить эту задачу. Просто сделай это".

Это прямой приказ, основанный на доверии к "образованию" нейросети.

Когда эта техника работает лучше всего?

Zero-Shot – это ваш основной инструмент для огромного спектра задач. Он максимально эффективен, в следующих случаях:

Первое! Задача является общеизвестной и стандартной. Например: перевод текста, написание краткого содержания, ответ на фактический вопрос, математические вычисления. Модель выполняла подобные задачи миллионы раз во время обучения и отлично знает, как это делать.

Второе! Вам нужна максимальная креативность. Когда вы хотите сгенерировать идеи, написать стихотворение или придумать варианты названий, вы часто не хотите ограничивать модель своими примерами. Вы хотите, чтобы она использовала всю широту своих "знаний" и выдала что-то действительно новое.

Третье! Вы хотите получить быстрый, "черновой" результат. Например, если вам нужно быстро набросать структуру статьи или получить **первоначальный черновик** текста, который вы затем будете самостоятельно редактировать и улучшать. В таких случаях нет смысла тратить время на подбор примеров, так как ваша цель – получить не финальный продукт, а лишь основу для него.

Примеры качественных Zero-Shot промтов!

Даже в рамках этой, казалось бы, простой техники есть огромная разница между плохим и хорошим промтом. Хороший Zero-Shot промт все равно использует компоненты, которые мы изучили: Роль, Контекст, Задачу и Ограничения.

Пример первый. Классификация текста!

Плохой промт:

Какой здесь тон? Отзыв: "Обслуживание было просто ужасным, ждал официанта 40 минут. Большие сюда ни ногой."

Модель, скорее всего, ответит правильно ("негативный"), но это слишком простой запрос.

Хороший промт:

Ты – аналитик клиентского опыта. Проанализируй следующий отзыв и классифицируй его по двум параметрам. Первый – тональность (позитивная, негативная или нейтральная). Второй – основная тема жалобы (сервис, качество еды, чистота, цена). Отзыв: "Еда была восхитительной, особенно понравился стейк. Но обслуживание было просто ужасным, ждал официанта 40 минут, а счетнесли еще полчаса. За такие деньги ожидаешь лучшего сервиса."

Здесь мы не даем примеров, но мы четко ставим задачу, задаем роль и формат. Это сложный, но все еще Zero-Shot промт, с которым справится хорошая модель.

Пример второй. Генерация идей!

Плохой промт:

Придумай название для кофейни.

Вы получите список из банальных и предсказуемых названий вроде "Кофейный уголок" или "Арома-кофе".

Хороший промт:

Ты – креативный директор брендингового агентства. Придумай 10 названий для новой кофейни в стиле лофт, расположенной в бывшем здании типографии. Целевая аудитория – творческая молодежь, фрилансеры. Названия должны быть короткими, современными и обыгрывать темы печати, букв, слов или бумаги.

Результаты будут на порядок интереснее.

Главное ограничение Zero-Shot.

Эта техника начинает давать сбои, когда ваша задача требует очень специфического, нестандартного формата ответа или уникального стиля, которому модель не была обучена. Если вы хотите, чтобы нейросеть отвечала в стиле вашей компании, классифицировала запросы по вашей внутренней системе или писала код по вашим внутренним стандартам, она не сможет сделать это без примеров. Она просто не знает, как это выглядит.

Именно в этот момент, когда силы прямого приказа оказывается недостаточно, на сцену выходят более точные и мощные техники.

Техника вторая! One-Shot.

One-Shot Prompting (промтинг "с одним выстрелом") – это техника, при которой вы даете модели ровно один при-

мер того, как нужно выполнить задачу, а затем просите ее применить этот образец к новым данным.

Суть техники: Вы предоставляете нейросети один эталонный пример в формате "Вот входные данные – Вот идеальный результат". Этот пример служит шаблоном. Модель смотрит на него, анализирует, какую трансформацию вы произвели, и пытается в точности скопировать эту трансформацию для вашего нового запроса.

Это невероятно эффективный способ быстро "объяснить" модели правила игры для конкретной, узкой задачи.

Когда эта техника работает лучше всего?

One-Shot – это ваш главный инструмент для форматирования и стилизации. Он идеален, в следующих случаях:

Первый. Вам нужен результат в строго определенном, но простом формате. Например, извлечь из текста данные и представить их в виде "Имя: , Фамилия: ".

Второй. Вы хотите задать простой, но специфичный стиль. Например, превращать длинные предложения в короткие и броские заголовки.

Третий. Задача достаточно проста, но ее сложно описать словами. Иногда проще один раз показать, чем сто раз объяснить.

Примеры качественных One-Shot промтов:

Пример первый. Трансформация формата данных.

Задача: У вас есть строки текста с информацией о пользователях, и вам нужно привести их к единому, чистому

формату.

Промт:

Извлеки данные из текста и представь их в формате 'Параметр: Значение', где каждый параметр находится на новой строке.

Пример: *Вход:* 'Заказ 951, пользователь Уильямс, сумма 250 долларов, статус в обработке.' *Выход:* Номер заказа: 951
Клиент: Уильямс Сумма: 250 долларов Статус: в обработке

Теперь обработай этот текст: *Вход:* 'Пользователь Джонсон, статус отправлен, id заказа 952, сумма 135.'

Модель видит ваш пример и точно понимает, что от нее требуется: найти соответствующие данные, присвоить им правильные названия (даже если порядок во входном тексте другой!) и расположить все на новых строках. Результат будет идеально отформатирован с первой попытки.

Пример второй. Имитация простого стиля!

Задача: Превращать формальные новостные поводы в короткие, интригующие заголовки для соцсетей.

Промт:

Преврати официальное утверждение в короткий, энергичный заголовок для новостного канала.

Пример: *Утверждение:* 'Согласно отчету аналитического отдела, наш новый продукт продемонстрировал рост квартальных продаж на 50 процентов.' *Заголовок:* 'Продажи взлетели: на пятьдесят процентов за квартал!'

Теперь преврати это утверждение: 'Интеграция но-

вого программного модуля в нашу систему позволила сократить среднее время обработки клиентских запросов на 25 процентов.'

Из одного примера модель поймет все правила: нужно сокращать, добавлять экспрессии ("взлетели") и ставить восклицательный знак.

Но что делать, если логика сложнее, чем простое копирование формата?

Техника третья: Few-Shot.

Few-Shot Prompting (промтинг "с несколькими выстрелами") – это самая мощная из трех техник. Здесь вы предоставляете модели не один, а несколько примеров (обычно от двух до пяти).

Суть техники: Когда модель видит несколько пар "вход - > выход", она не просто копирует формат. Она начинает **анализировать и выявлять скрытую закономерность**. Она пытается понять логику, которая связывает все ваши примеры. По сути, вы не даете ей шаблон, вы даете ей мини-урок, проводите экспресс-обучение прямо внутри промта.

Это позволяет решать гораздо более сложные и неочевидные задачи.

Когда эта техника работает лучше всего?

Few-Shot незаменим, в следующих случаях:

Первый! Задача требует сложной классификации по вашим собственным правилам. Например, распределить отзывы клиентов по категориям, которые придумали

вы, а не модель.

Второй! Логика трансформации данных неочевидна и ее трудно описать словами.

Третий! Вам нужен очень специфичный стиль или тон, который имеет несколько особенностей.

Примеры качественных Few-Shot промтов:

Пример первый. Сложная классификация!

Задача: Вы работаете в поддержке и вам нужно автоматически распределять входящие сообщения от пользователей по трем внутренним категориям: "Проблема с оплатой", "Технический вопрос" и "Предложение по улучшению".

Промт:

Классифицируй сообщение пользователя по одной из трех категорий: "Проблема с оплатой", "Технический вопрос", "Предложение по улучшению". Вот несколько примеров:

Пример 1: *Сообщение:* 'Здравствуйте, не могу найти, где отменить подписку в настройках.' *Категория:* 'Технический вопрос'

Пример 2: *Сообщение:* 'С меня списали деньги за этот месяц дважды! Разберитесь!' *Категория:* 'Проблема с оплатой'

Пример 3: *Сообщение:* 'Было бы очень здорово, если бы в приложении появилась темная тема.' *Категория:* 'Предложение по улучшению'

Пример 4: *Сообщение:* 'После последнего обновления

программа стала вылетать при запуске.' Категория: 'Технический вопрос'

Теперь классифицируй это сообщение: *Сообщение:* 'Почему я не могу оплатить подписку картой вашего банка-партнера?'

Посмотрите, насколько сложна логика. В последнем сообщении есть и слово "оплатить", и слово "банк", но по сути это техническая проблема, связанная с работой платежного шлюза, а не проблема списания. Описать это правило словами было бы очень сложно. Но благодаря нескольким примерам, модель улавливает эту закономерность и с высокой вероятностью даст правильный ответ: "Технический вопрос".

Итог: Когда какую технику выбрать?

Теперь у вас есть три мощных инструмента. Вот простая ментальная модель, которая поможет вам быстро выбирать нужный:

Первое! Если Задача стандартная, и модель точно знает, как ее делать?"

Используйте **Zero-Shot**. Это самый быстрый и эффективный способ для перевода, суммаризации, мозгового штурма и других общих задач. Не усложняйте без необходимости.

Второе! Если вам нужно, чтобы ответ был в очень специфическом, но простом формате или стиле?"

Используйте **One-Shot**. Один хороший пример – лучший способ показать модели требуемый шаблон, будь то форматирование текста или имитация простого стиля.

Третье! Если задача сложная, а логика ее выполнения неочевидна и вам нужно, чтобы модель поняла сложные, негласные правила?"

Используйте **Few-Shot**. Это ваш главный инструмент для сложных классификаций, извлечения данных по вашим личным правилам и обучения модели нюансам вашего уникального стиля.

Освоив эти три фундаментальных способа постановки задач, вы получаете ключ к управлению нейросетью на высоком уровне. Вы больше не действуете вслепую – вы осознанно выбираете стратегию в зависимости от сложности и специфики вашей цели.

Глава 6: Продвинутые стратегии. Chain-of-Thought, Self-Consistency и другие.

В предыдущей главе мы освоили три основных способа постановки задач. Вы научились осознанно выбирать между прямым приказом (**Zero-Shot**), предоставлением шаблона (**One-Shot**) и обучением на закономерностях (**Few-Shot**). Это дало вам ключ к управлению нейросетью на высоком уровне.

Теперь мы поднимемся еще на одну ступень. Мы поговорим о стратегиях, которые нужны тогда, когда задача требует не просто ответа, а **рассуждений**. Большинство ошибок нейросети совершают не потому, что не знают ответа, а потому, что "торопятся" и пытаются выдать его одним скачком, пропуская промежуточные логические шаги. Продвинутые стратегии как раз и направлены на то, чтобы заставить мо-

дель не торопиться, а думать последовательно.

Chain-of-Thought: Заставляем нейросеть "думать вслух"

Chain-of-Thought, или "Цепочка Мыслей" – это, возможно, самый важный прорыв в промт-инжиниринге за последние годы. Это простая, но невероятно мощная техника, которая кардинально повышает точность ответов модели в задачах, требующих логики, математических вычислений или анализа последовательности действий.

Суть техники: Вы прямо просите нейросеть не просто дать финальный ответ, а сначала **расписать всю цепочку своих рассуждений шаг за шагом**.

Почему это работает? Вспомним, что нейросеть – это система, предсказывающая следующий токен. Когда вы просите ее решить задачу и сразу дать ответ, ей нужно совершить огромный "прыжок" к финальному набору токенов, и вероятность ошибки в этом прыжке очень велика. Но когда вы просите ее расписать решение по шагам, каждая следующая мысль – это маленький, логичный и гораздо более вероятный шаг от предыдущей. Вы заставляете ее решать задачу последовательно, как это делал бы человек, и это резко снижает количество ошибок. По сути, вы просите ее "показать свою работу".

Давайте посмотрим на элементарном примере, где стандартный подход часто дает сбой.

Задача: Простая логическая задачка на внимательность.

Плохой промт (стандартный Zero-Shot):

На столе лежит 5 яблок. Я взял 2 яблока, а затем моя сестра положила на стол еще 3. После этого мы съели 1 яблоко. Сколько яблок осталось на столе?

Современные модели, скорее всего, справятся с этой задачей. Но еще совсем недавно многие из них могли ошибиться, запутавшись в последовательности действий (например, сначала сложить $5+3$, а потом вычесть 2 и 1). Они пытались выдать результат сразу.

Теперь давайте применим магию Chain-of-Thought.

Хороший промт (с использованием Chain-of-Thought):

На столе лежит 5 яблок. Я взял 2 яблока, а затем моя сестра положила на стол еще 3. После этого мы съели 1 яблоко. Сколько яблок осталось на столе? **Распиши свои рассуждения по шагам, а в конце дай финальный ответ.**

Одна простая фраза в конце меняет все.

Вероятный результат:

Конечно, давайте разберем по шагам:

Начальное состояние: На столе было 5 яблок.

Действие 1: Вы взяли 2 яблока. Значит, на столе осталось: $5 \text{ минус } 2 \text{ равно } 3$ яблока.

Действие 2: Ваша сестра положила еще 3 яблока. Теперь на столе: $3 \text{ плюс } 3 \text{ равно } 6$ яблок.

Действие 3: Вы съели 1 яблоко. В итоге на столе осталось: $6 \text{ минус } 1 \text{ равно } 5$ яблок.

Финальный ответ: На столе осталось 5 яблок.

Что мы получили?

Точность. Вероятность правильного ответа приблизилась к ста процентам.

Прозрачность. Вы видите всю логику модели. Если бы она ошиблась на каком-то шаге, вы бы это сразу заметили и могли бы ее поправить.

Управляемость. Вы превратили сложную задачу в последовательность простых, контролируемых операций.

Когда еще необходимо использовать Chain-of-Thought?

Эта техника – ваш главный помощник в любых задачах, где важен не только результат, но и процесс его получения:

Решение любых математических и логических задач.

Анализ текста. Например: "Прочитай этот договор и по шагам объясни, какие пункты могут быть рискованными для арендатора".

Составление планов и инструкций. Например: "Составь пошаговый план по организации переезда в новый офис".

Отладка кода. Например: "Проанализируй этот код и по шагам объясни, где может возникать ошибка".

Техника Chain-of-Thought – это ваш первый и самый важный шаг от простых команд к управлению "мышлением" нейросети. Одна эта техника способна кардинально изме-

нить качество вашей работы со сложными задачами.

Теперь возникает следующий вопрос. А что, если существует несколько путей рассуждений? И как выбрать из них самый надежный? В этом нам поможет техника Self-Consistency.

Self-Consistency! Мнение большинства.

Техника Chain-of-Thought значительно повышает надежность ответов. Но что, если задача настолько сложна, что даже пошаговые рассуждения могут привести к ошибке? Иногда существует несколько путей решения, и некоторые из них могут быть неверными. Здесь нам на помощь приходит стратегия **Self-Consistency** (Самосогласованность).

Суть техники: Эта техника является надстройкой над Chain-of-Thought. Вместо того, чтобы просить модель сгенерировать одну цепочку рассуждений, вы просите ее сделать это **несколько раз**, возможно, даже попросив "подумать над задачей по-разному". Затем вы смотрите на финальные ответы, полученные в результате каждой цепочки, и выбираете тот, который встречается чаще всего.

Почему это работает? Это принцип "мнения большинства", примененный к одной нейросети. Представьте, что вы дали сложную задачу комитету из пяти независимых экспертов. Один из них может ошибиться в расчетах, другой – неверно истолковать условие. Но скорее всего, большинство придет к правильному ответу. Генерируя несколько независимых цепочек рассуждений, вы многократно снижаете

вероятность случайной ошибки. Даже если одна из "мыслительных" попыток модели окажется неверной, остальные, скорее всего, приведут к правильному результату.

Давайте рассмотрим на примере, который чуть сложнее предыдущего.

Задача: Логическая задачка с процентами и фиксированной скидкой, где важен порядок действий.

Промт с использованием Self-Consistency:

Цена куртки – 250 долларов. На нее действует сезонная скидка 20%. Кроме того, у вас есть персональный купон на скидку в 25 долларов. Какова итоговая цена куртки? **Реши эту задачу тремя разными способами, расписав рассуждения для каждого. В конце сравни результаты и дай окончательный ответ.**

Такой промт запускает механизм Self-Consistency. Модель попытается найти разные подходы, и здесь есть вероятность, что один из них окажется неверным.

Вероятный результат:

Способ 1. Сначала процентная скидка, потом фиксированная.

Начальная цена: 250 долларов.

Рассчитываем скидку 20%: $250 \times 0.2 = 50$ долларов.

Цена после процентной скидки: $250 - 50 = 200$ долларов.

Применяем купон на 25 долларов: $200 - 25 = 175$ долларов. *Промежуточный ответ: 175 долларов.*

Способ 2. Расчет остаточной стоимости.

Начальная цена: 250 долларов.

Скидка 20% означает, что от цены остается 80%.

Рассчитываем остаточную стоимость: 250 умножить на $0.8 = 200$ долларов.

Отнимаем фиксированную скидку по купону: $200 - 25 = 175$ долларов. *Промежуточный ответ: 175 долларов.*

Способ 3. Ошибочный путь.

Начальная цена: 250 долларов.

Сначала применим фиксированную скидку по купону: $250 - 25 = 225$ долларов.

Теперь от этой новой суммы рассчитаем скидку 20%: 225 умножить на $0.2 = 45$ долларов.

Финальная цена: $225 - 45 = 180$ долларов. *Промежуточный ответ: 180 долларов.*

Сравнение и финальный ответ: Сравнение показывает, что два из трех способов рассуждений привели к ответу **175 долларов**, и только один – к ответу **180 долларов**. Метод большинства указывает, что наиболее вероятный и правильный ответ – 175. **Окончательный ответ:** Итоговая цена куртки – 175 долларов.

Этот пример наглядно показывает всю мощь Self-Consistency. Модель способна сгенерировать правдоподобную, но неверную цепочку рассуждений (сначала применив купон). Однако "голосование" нескольких независимых подходов позволяет отфильтровать эту ошибку и прийти к пра-

вильному выводу. Это ваша самая надежная страховка от случайных ошибок в критически важных задачах.

Decomposition!

Последняя стратегия в нашем списке – **Decomposition** (Декомпозиция). Это не столько техника промтинга, сколько стратегический подход к решению больших и комплексных задач.

Суть техники: Вместо того, чтобы пытаться решить одну огромную задачу одним гигантским промтом, вы **вручную разбиваете ее на несколько последовательных, логически связанных подзадач** и решаете их шаг за шагом в режиме диалога.

Почему это работает? Этот подход решает сразу две проблемы. Во-первых, он помогает справиться с ограничением окна контекста, так как на каждом шаге модель фокусируется только на одной, относительно небольшой задаче. Во-вторых, он снижает "когнитивную нагрузку" на модель. Пытаясь сделать десять вещей одновременно, нейросеть, скорее всего, сделает каждую из них посредственно. Но если вы попросите ее сделать десять вещей по очереди, она выполнит каждую на высшем уровне.

При этом подходе вы выступаете в роли **проектного менеджера**, а нейросеть – в роли вашего высококвалифицированного исполнителя.

Пример. Создание контента "под ключ".

Задача: Написать экспертную статью в блог о влиянии

кофе на продуктивность, найти под нее релевантные научные исследования и придумать заголовки для анонса в соцсетях.

Плохой промт (все в одном):

Напиши длинную экспертную статью о плюсах и минусах кофе для продуктивности. Обязательно сошлись на 2-3 научных исследования. В конце предложи 5 ярких заголовков для анонса этой статьи в телеграм-канале.

Скорее всего, вы получите среднюю по качеству статью, выдуманные "исследования" и банальные заголовки. Задача слишком комплексна для одного промта.

Хороший подход (Декомпозиция на шаги):

Шаг первый (Поиск информации). "Мы готовим статью о влиянии кофе на продуктивность. Найди, пожалуйста, названия и краткие выводы трех реальных научных исследований (с указанием университетов или институтов), которые изучали связь между потреблением кофеина и когнитивными функциями, такими как внимание и память."

Шаг второй (Составление плана). "Отлично, спасибо за исследования. Теперь, опираясь на них и на общие знания, составь подробный план для экспертной статьи на эту тему. План должен включать введение, три основных раздела (например, 'Краткосрочные эффекты', 'Долгосрочные риски', 'Как пить кофе правильно') и заключение."

Шаг третий (Написание текста по частям). "План утвержден. Давай начнем писать. Напиши, пожалуйста, вве-

дение и первый раздел ('Краткосрочные эффекты'), используя экспертный, но понятный язык."

(...и так далее, пока вся статья не будет написана...)

Шаг четвёртый (Генерация заголовков). "Статья готова и выглядит отлично. Теперь, на основе финального текста, придумай 5 ярких и интригующих заголовков для анонса этой статьи в нашем телеграм-канале."

Такой подход требует больше вашего участия, но результат будет несравнимо выше по качеству. Вы получаете полный контроль над каждым этапом, можете моментально вносить правки и в итоге собираете из качественных частей идеальный финальный продукт.

Заключение.

Итак, теперь в вашем арсенале есть три мощные стратегии для работы со сложными задачами:

Chain-of-Thought – чтобы заставить модель рассуждать и сделать ее "мысли" прозрачными.

Self-Consistency – чтобы перепроверить результат через "мнение большинства" и повысить надежность.

Decomposition – чтобы управлять большими проектами, разбивая их на выполнимые шаги.

Мы изучили фундаментальные и продвинутые стратегии, которые применимы к работе с любыми нейросетями. Теперь, вооружившись этим мощным инструментарием, мы готовы погрузиться в специфику работы с разными типами моделей. В следующей большой части нашей книги мы подроб-

но разберем, как применять все эти знания для генерации первоклассных текстов для любых целей.

Часть II: Промт инжиниринг для текстовых моделей.

В первой части мы заложили прочный теоретический фундамент. Вы поняли, как работают нейросети, научились конструировать промты из шести ключевых компонентов и освоили универсальные стратегии – от базовых техник до продвинутых. Теперь мы переходим к самому главному: к применению всех этих знаний для решения конкретных, реальных задач в одной из самых востребованных сфер – работе с текстом.

Эта часть книги – ваш практический гид по созданию текстового контента любого типа и сложности. Мы пройдем путь от разработки структуры статьи до написания больших, многостраничных материалов и сценариев. Умение за один час выполнить работу, на которую раньше уходил целый день, – это и есть результат применения правильных технологий, которые мы и будем здесь осваивать.

Глава 7. Создание контента. Статьи, посты для блогов и сценарии.

Самая частая ошибка, которая ведет к получению посредственного текста, – это попытка создать его "в одно действие". Промт в стиле "Напиши статью о..." – это не серьёзный подход к делу. Вы отдаете 90% контроля над результатом на усмотрение модели.

Ключевой принцип профессиональной работы с контентом: **любая серьезная работа над текстом начинается с разработки его структуры**. Тщательно продуманный план – это фундамент, который гарантирует логику, последовательность и полноту раскрытия темы. Использование нейросети для создания этого плана – первый и самый важный шаг в любом контентном проекте. Для этого мы будем использовать уже знакомую нам стратегию **Декомпозиции**, разделив большую задачу ("написать статью") на два этапа: сначала "создать план", и только потом "написать текст по плану".

Давайте разберем сам процесс создания промта для разработки структуры.

Задача: Подготовить подробную статью для блога на тему "Как новичку выбрать свою первую акустическую гитару".

Как спроектировать промт для создания плана:

Ваша цель – не просто попросить план, а дать модели четкое техническое задание на его разработку. Ваш промт должен включать в себя несколько ключевых инструкций:

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.