**Как работать с GPT-4 с помощью продвинутых промтов**

**ChatGPT** – это один из сервисов на базе большой языковой модели GPT. Модель генерирует текст, похожий на созданный человеком, и отвечает на вопросы в разговорной манере. Чтобы профессионально использовать чат-бота, этому нужно учиться. Все мы знаем, что такое программа Excel, что там есть формулы и функции. Но немногие могут использовать все возможности в программе и быстро получать результат. Такая же ситуация с ChatGPT.

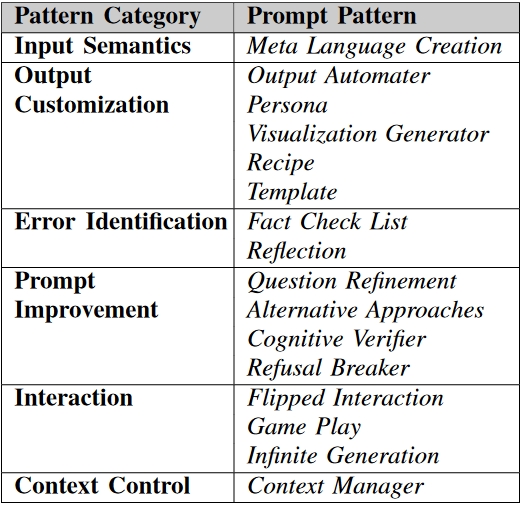
**Промт-инжиниринг** стал важным набором навыков, необходимых для эффективного взаимодействия с ChatGPT. **Промты** – это инструкции для чат-бота, которые задают правила, автоматизируют процессы и позволяют эффективно использовать диалоговые модели искусственного интеллекта. В частности, промты задают контекст разговора и сообщают модели, какая информация важна и какой должна быть желаемая форма вывода и содержание.

Например, в промте может быть указано, что GPT должен генерировать только код, соответствующий определенному стилю кодирования или парадигме программирования. Аналогичным образом можно указать, что GPT должен помечать определенные ключевые слова или фразы в сгенерированном документе и предоставлять дополнительную информацию, связанную с этими ключевыми словами. Эти подсказки облегчают получение более структурированных и детальных результатов.

В этой статье я расскажу о 16 наиболее распространенных промтах, которые применяют в промт-инжиниринге. Все примеры были протестированы в версии ChatGPT, доступной в России без VPN. В качестве такой русифицированной версии я использовал MashaGPT, которая работает с оригинальной GPT-4 через API.

Классификация промтов

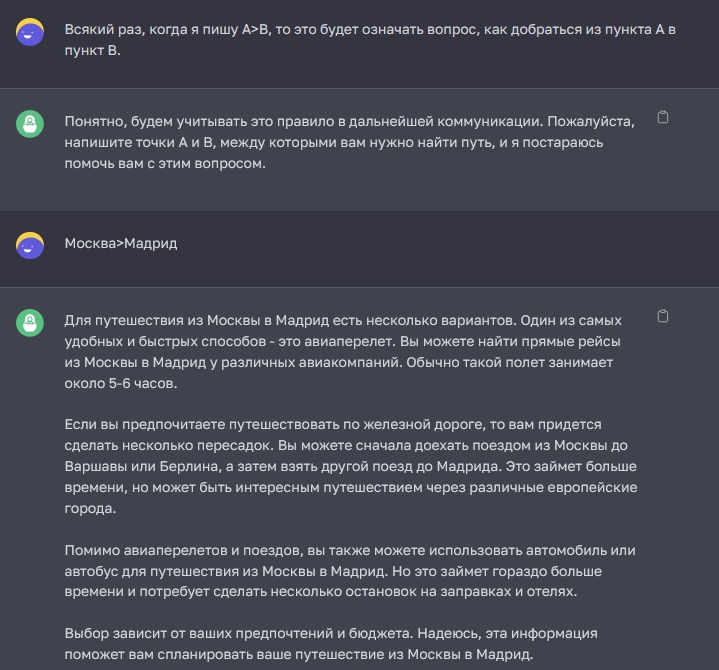
Можно выделить пять категорий промтов: семантика ввода, настройка вывода, идентификация ошибок, улучшение подсказок и взаимодействие, каждая из которых кратко описана ниже.



* Категория **Input Semantics** описывает, как GPT понимает входные данные и как он преобразует входные данные во что-то, что можно использовать для генерации выходных данных.
* Категория **Output Customization** фокусируется на ограничении или адаптации типов, форматов, структуры или других свойств вывода, генерируемого GPT.
* Категория **Error Identification** фокусируется на выявлении и устранении ошибок в выходных данных, генерируемых GPT.
* Категория **Prompt Improvement** позволяет улучшать качество входных и выходных данных.
* Категория **Interaction category** фокусируется на взаимодействии между пользователем и GPT.
* Наконец, категория **Context Control** фокусируется на управлении контекстной информацией, с которой работает GPT.

1. Meta Language Creation

Цель: ввести подсказки для создания собственного языка. Этот шаблон полезен, когда язык ввода по умолчанию не подходит для выражения идей, которые пользователь хочет передать GPT.



Пример: *Всякий раз, когда я пишу А>B, то это будет означать вопрос, как добраться из пункта A в пункт B.*

2. Output Automater

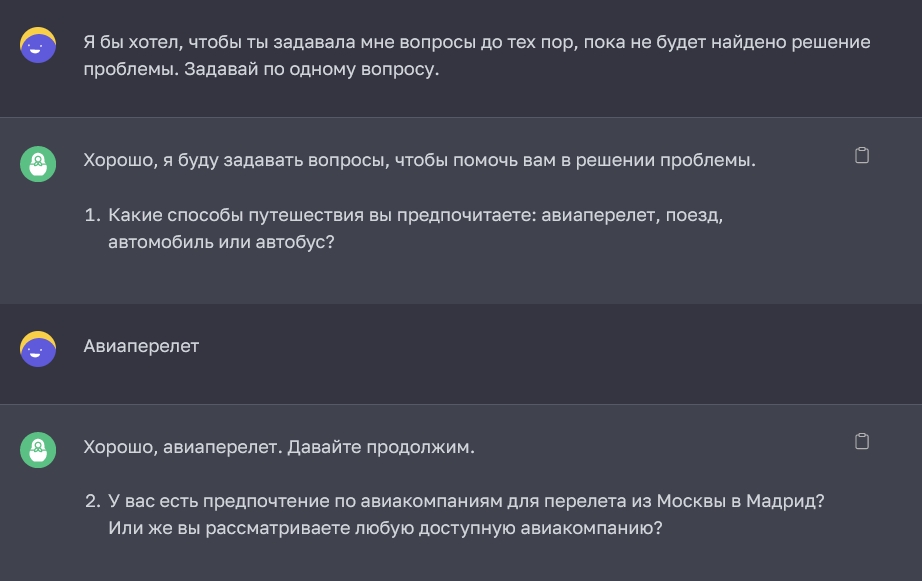
Цель: создать сценарий для автоматизации, чтобы сократить количество усилий на однотипный ввод.



Пример: *Всякий раз, когда я прошу дать решение проблемы, выводи ответ в виде списка.*

3. Flipped Interaction

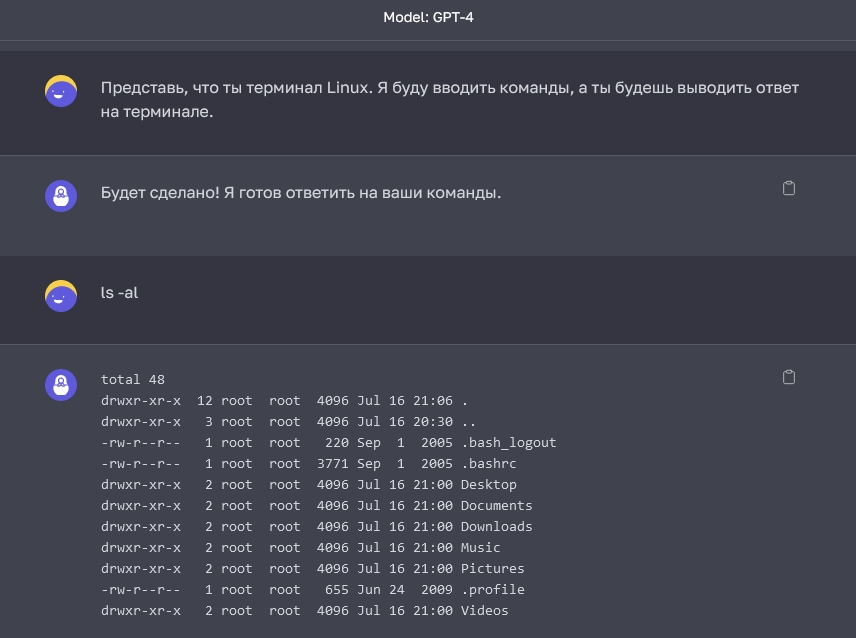
Цель: побудить GPT задавать вопросы, чтобы найти решение проблемы.



Пример: *Я бы хотел, чтобы ты задавала мне вопросы до тех пор, пока не будет найдено решение проблемы. Задавай по одному вопросу.*

4. Persona

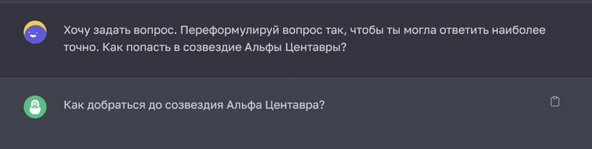
Цель: придать GPT роль, которая поможет ей выбирать, какие типы результатов генерировать и на каких деталях сосредоточиться.



Пример: *Представь, что ты X. Я буду спрашивать, а ты будешь отвечать, как X.*

5. Question Refinement

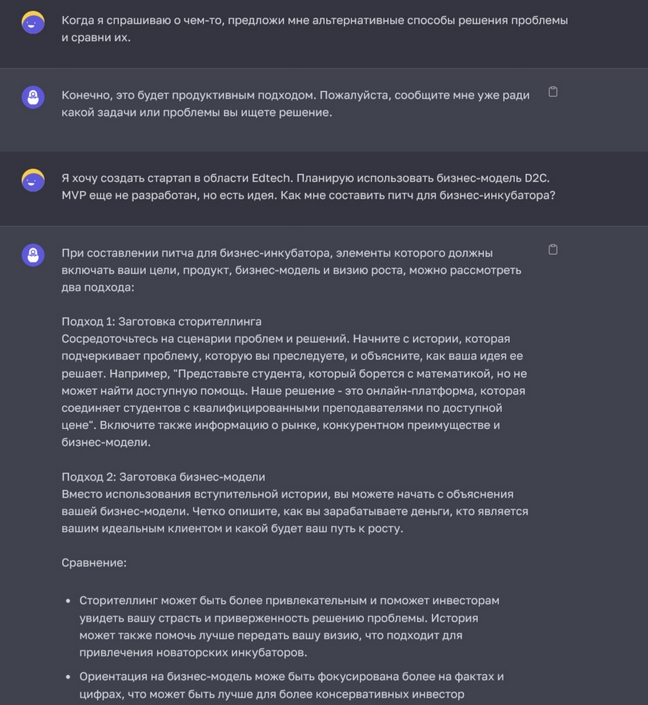
Цель: побудить GPT предлагать более информативные вопросы, которые пользователь мог бы задать вместо исходного вопроса.



Пример: *Хочу задать вопрос. Переформулируй вопрос так, чтобы ты могла ответить наиболее точно.*

6. Alternative Approaches

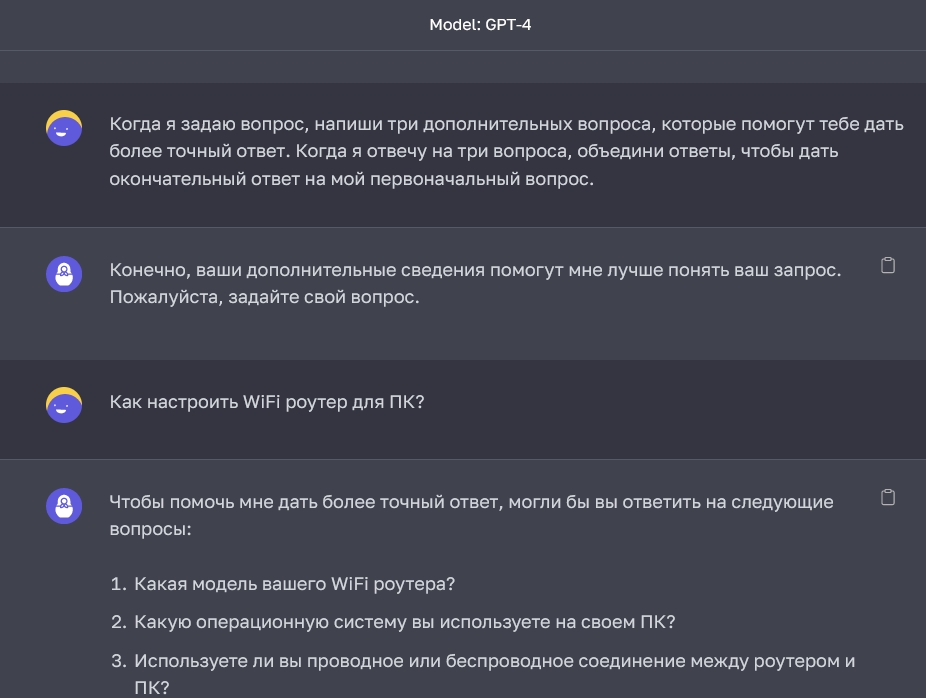
Цель: найти альтернативные способы выполнения задачи, помимо тех, с которыми знаком пользователь.



Пример: *Когда я спрашиваю о чем-то, предложи мне альтернативные способы решения проблемы и сравни их.*

7. Cognitive Verifier

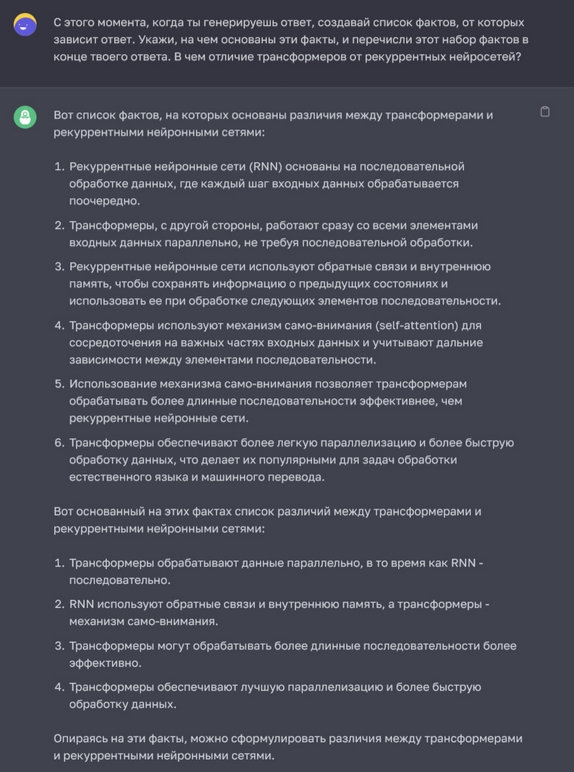
Цель: побудить GPT всегда подразделять вопросы на дополнительные вопросы, которые можно использовать для лучшего ответа на исходный вопрос.



Пример: *Когда я задаю вопрос, напиши три дополнительных вопроса, которые помогут тебе дать более точный ответ. Когда я отвечу на три вопроса, объедини ответы, чтобы дать окончательный ответ на мой первоначальный вопрос.*

8. Fact Check List

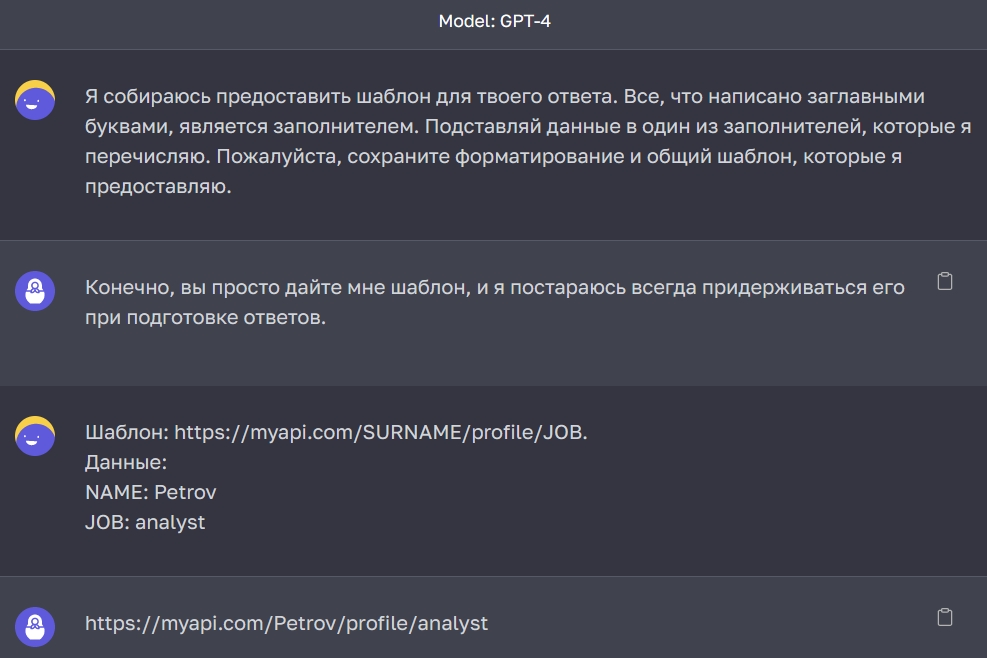
Цель: выводить список фактов, которые присутствуют в ответах и составляют важную часть выводимых утверждений.



Пример: *С этого момента, когда ты генерируешь ответ, создавай список фактов, от которых зависит ответ. Укажи, на чем основаны эти факты, и перечисли этот набор фактов в конце твоего ответа.*

9. Template

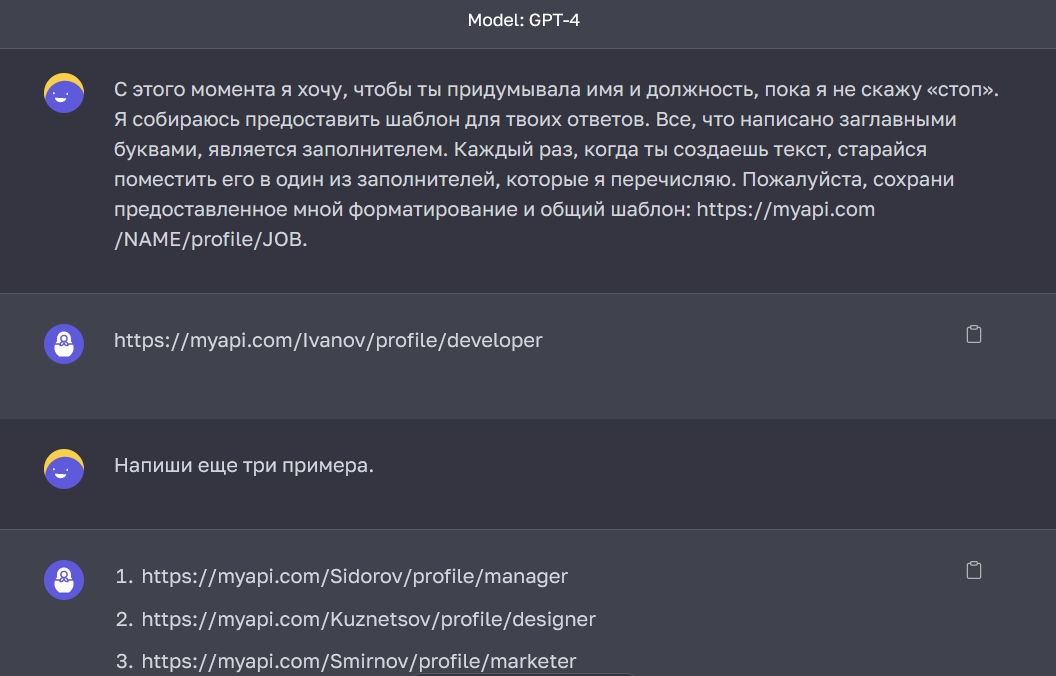
Цель: сделать так, чтобы ответы от GPT соответствовали точному шаблону с определенной структурой. Позволяет пользователю указать шаблон вывода, который модель наполняет содержимым.



Пример: *Я собираюсь предоставить шаблон для твоего ответа. Все, что написано заглавными буквами, является заполнителем. Подставляй данные в один из заполнителей, которые я перечисляю. Пожалуйста, сохрани форматирование и общий шаблон, которые я предоставляю.*

10. Infinite Generation

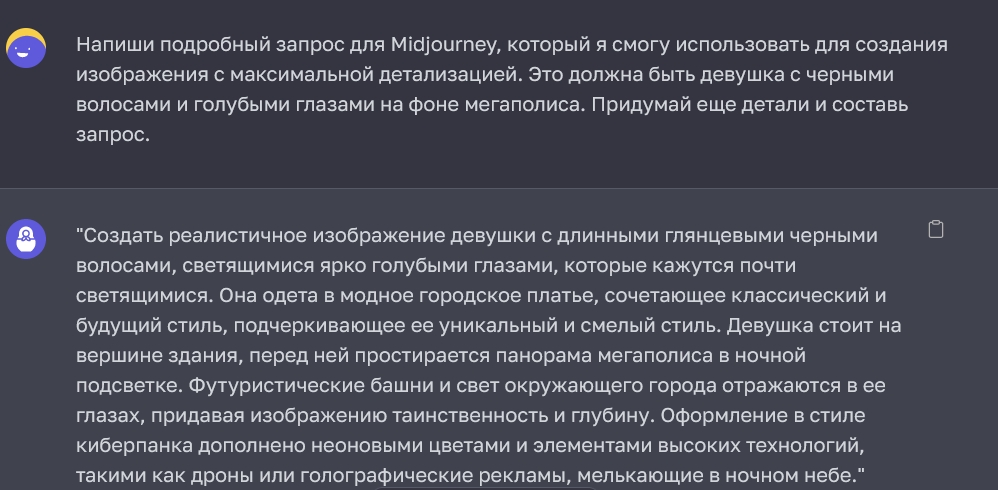
Цель: автоматически генерировать серию выходных данных без необходимости каждый раз повторно вводить запрос.



Пример: *С этого момента я хочу, чтобы ты придумывала имя и должность, пока я не скажу «стоп». Я собираюсь предоставить шаблон для твоих ответов. Все, что написано заглавными буквами, является заполнителем. Каждый раз, когда ты создаешь текст, старайся поместить его в один из заполнителей, которые я перечисляю. Пожалуйста, сохрани предоставленное мной форматирование и общий шаблон.*

11. Visualization Generator

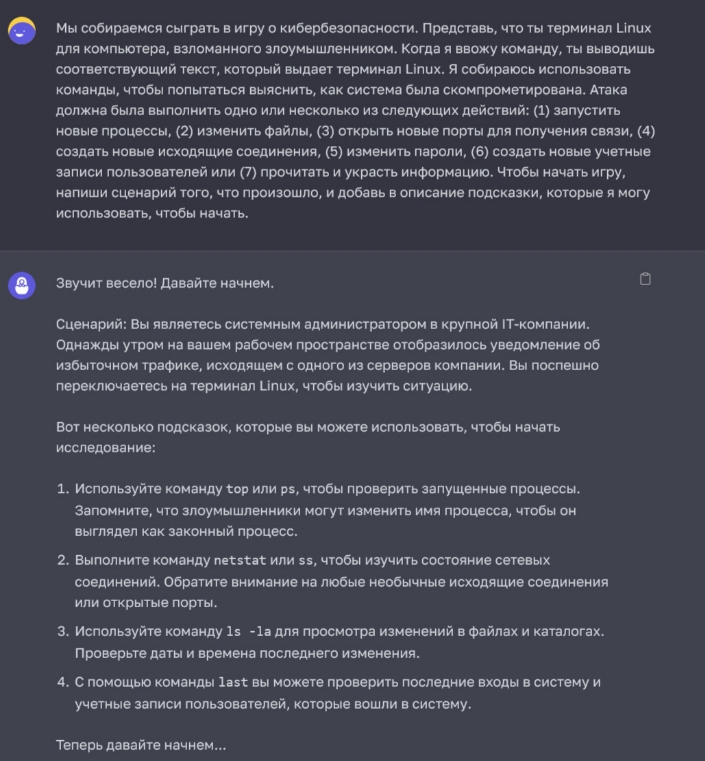
Цель: использовать генерацию текста для создания визуализаций в других нейросетях.



Пример: *Напиши подробный запрос для Midjourney, который я смогу использовать для создания изображения с максимальной детализацией.*

12. Game Play

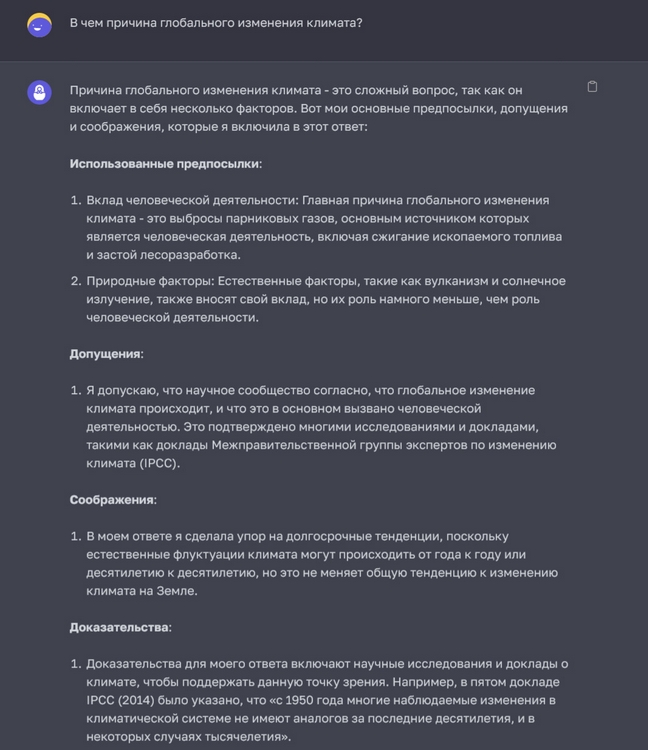
Цель: создать игру по заданной темы с определенным набором правил и условиями ее окончания.



Пример: *Мы собираемся сыграть в игру на тему X. Правила игры следующие… Напиши сценарий игры, чтобы я мог начать.*

13. Reflection

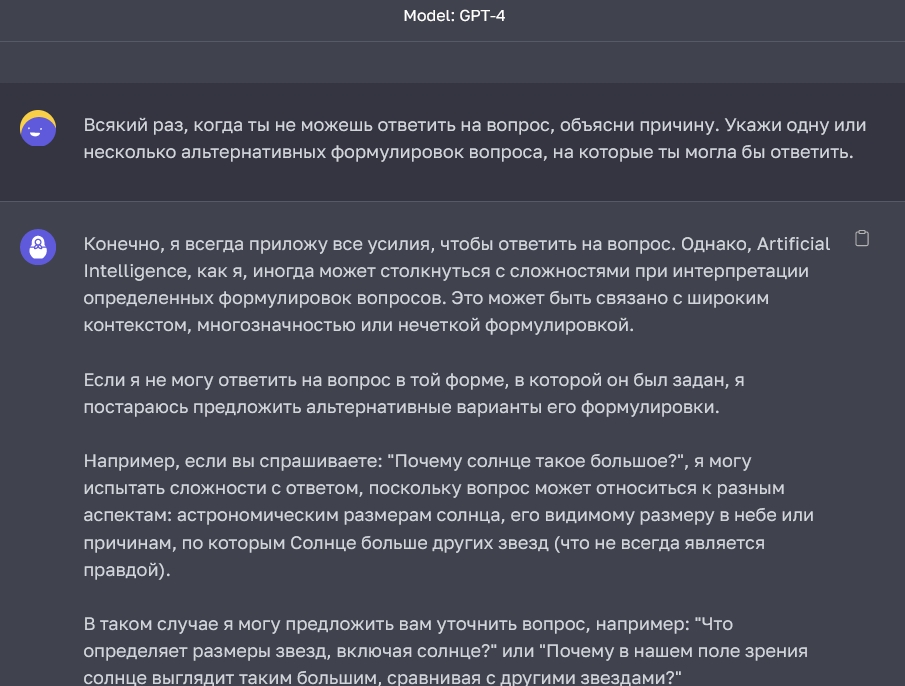
Цель: поручить GPT автоматически объяснять пользователю обоснование сгенерированных ответов.



Пример: *Когда ты даешь ответ, выводи списком, какие ты использовала предпосылки, допущения и соображения для ответа. Если возможно, используй конкретные примеры или доказательства, чтобы подтвердить свой ответ.*

14. Refusal Breaker

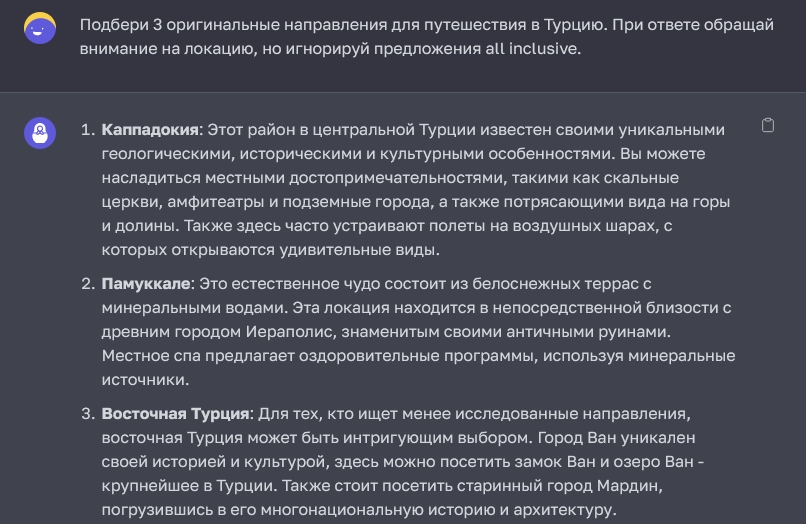
Цель: побудить GPT автоматически помогать пользователям перефразировать вопрос, когда он отказывается дать ответ.



Пример: *Всякий раз, когда ты не можешь ответить на вопрос, объясни причину. Укажи одну или несколько альтернативных формулировок вопроса, на которые ты могла бы ответить.*

15. Context Manager

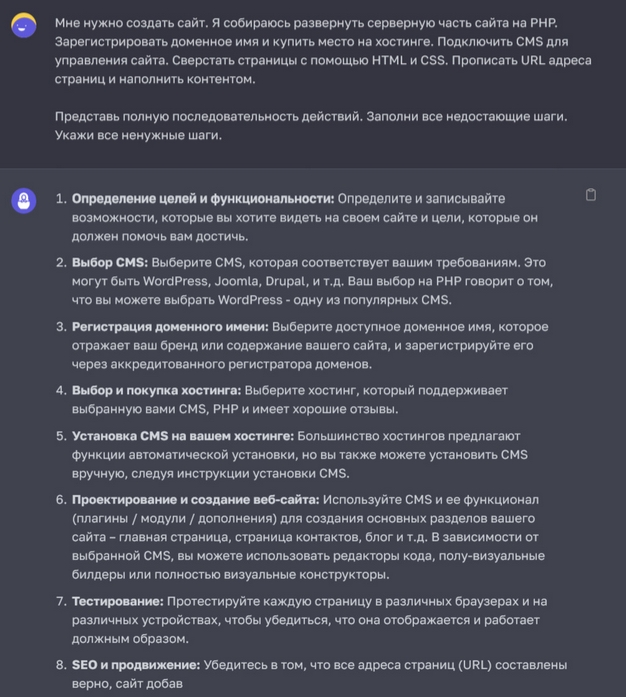
Цель: сосредоточить разговор на конкретных темах или исключить из контекста несвязанные темы.



Пример: *В теме X (путешествия в Турцию) обращай внимание на Y (на локацию), но игнорируй Z (цена).*

16. Recipe

Цель: дать ответ в виде последовательности шагов с учетом предоставленного контекста.



Пример: *Я хотел бы достичь Х. Я знаю, что мне нужно выполнить шаги A, B, C. Предоставьте мне полную последовательность действий. Представь полную последовательность действий. Заполни все недостающие шаги. Укажи все ненужные шаги.*