ЛЕКЦИЯ 13

PROMPT ENGINEERING: МЕТОДЫ ПРОЕКТИРОВАНИЯ ЭФФЕКТИВНЫХ ЗАПРОСОВ

Основные понятия • определения • схемы • таблицы • графики • примеры

Преподаватель: проф.Ахмедиярова А.Т.

Структура и основные разделы

01 Понятие Prompt Engineering

02 Почему формулировка запроса влияет на результат

03 Основные компоненты эффективного промпта

04 Конкретность и однозначность

05 Контекст в промпте

06 Роль и аудитория

07 Формат результата

08 Zero-Shot prompting

09 One-Shot и Few-Shot prompting

10 Как выбирать примеры для Few-Shot

11 Структурированное рассуждение

12 Декомпозиция сложной задачи

Понятие Prompt Engineering

Prompt — запрос, инструкция или совокупность входных данных, передаваемых модели искусственного интеллекта. Prompt Engineering — процесс проектирования, проверки и улучшения таких запросов для получения результата, соответствующего поставленной задаче.

Ключевая идея: Prompt — запрос, инструкция или совокупность входных данных, передаваемых модели искусственного интеллекта. Prompt Engineerin

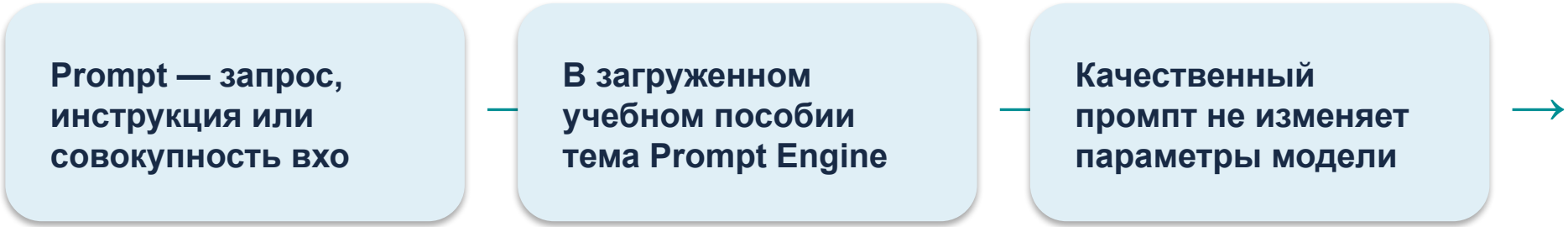
Понятие Prompt Engineering — продолжение

В загруженном учебном пособии тема Prompt Engineering включает принципы создания промптов, Zero-Shot, Few-Shot, Chain-of-Thought и практические кейсы. В силлабусе также предусмотрено освоение принципов Prompt Engineering и эффективного взаимодействия с современными моделями искусственного интеллекта. Качественный промпт не изменяет параметры модели.

Понятие Prompt Engineering — продолжение

Он организует информацию в контексте и уточняет, что именно требуется сделать, с какими данными, при каких ограничениях и в каком формате.

Схема: Понятие Prompt Engineering



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Почему формулировка запроса влияет на результат

Большая языковая модель генерирует ответ на основе доступного контекста. Если задача сформулирована слишком широко, модель вынуждена самостоятельно предполагать недостающие условия. Это повышает вероятность нерелевантного, слишком общего или неправильно оформленного ответа. Сравним два запроса. Первый: «Проанализируй данные».

Ключевая идея: Большая языковая модель генерирует ответ на основе доступного контекста. Если задача сформулирована слишком широко, модель вы

Почему формулировка запроса влияет на результат — продолжение

Второй: «Проанализируй таблицу результатов классификации, сравни precision, recall и F1-score для трёх моделей и сформулируй вывод в пяти предложениях без добавления данных, отсутствующих в таблице». Во втором случае задача, объект анализа, критерии и формат определены значительно точнее.

Сравнительная таблица: Почему формулировка запроса влияет на результат

Почему формулировка запроса влияет на результат	Большая языковая модель генерирует ответ на основе доступного контекста. Если задача сформулирована слишком широко,
Основные компоненты эффективного промпта	Сравним два запроса. Первый: «Проанализируй данные». Второй: «Проанализируй таблицу результатов классификации, срав

Основные компоненты эффективного промпта

Рисунок 1 – Основные компоненты эффективного промпта

Конкретность и однозначность

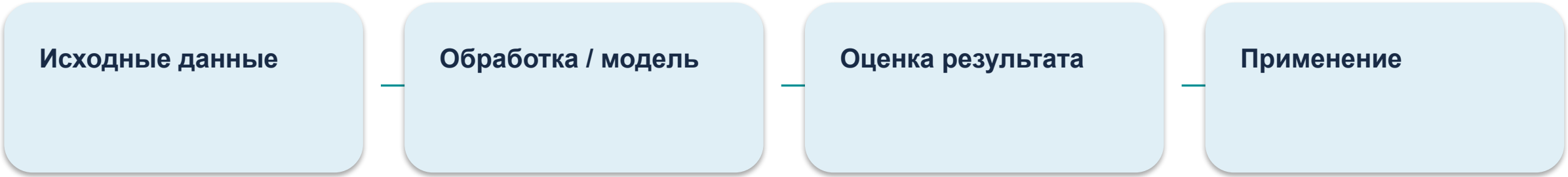
Промпт должен уменьшать неопределённость задачи. Формулировки «сделай хорошо», «проанализируй подробно» или «напиши правильно» не задают измеримого результата. Лучше определить объём, аудиторию, структуру, критерии и ограничения. Например, вместо «Объясни нейронные сети» можно написать: «Объясни принцип работы многослойного персептрона магистрантам первого года обучения.

Ключевая идея: Промпт должен уменьшать неопределённость задачи. Формулировки «сделай хорошо», «проанализируй подробно» или «напиши правильно

Конкретность и однозначность — продолжение

Используй определения входного, скрытого и выходного слоёв, приведите один числовой пример и завершите сравнением MLP и логистической регрессии».

Схема: Конкретность и однозначность



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Контекст в промпте

Контекст — информация, необходимая модели для выполнения задачи. Это может быть исходный текст, таблица, описание предметной области, требования к документу или ранее принятые обозначения. Если ответ должен основываться на конкретном документе, целесообразно явно ограничить источник: «Используй только предоставленный фрагмент. Если ответа в нём нет, укажи, что информации недостаточно».

Ключевая идея: Контекст — информация, необходимая модели для выполнения задачи. Это может быть исходный текст, таблица, описание предметной

Контекст в промпте — продолжение

Такое ограничение не гарантирует отсутствие ошибки, но делает задачу более определённой.

Роль и аудитория

В промпте можно указать профессиональный контекст или целевую аудиторию. Это помогает выбрать терминологию, глубину объяснения и форму ответа. Указание роли не даёт модели дополнительных полномочий или гарантированной экспертности. Это прежде всего способ задать стиль и перспективу ответа.

Ключевая идея: В промпте можно указать профессиональный контекст или целевую аудиторию. Это помогает выбрать терминологию, глубину объяснени

Сравнительная таблица: Роль и аудитория

Роль и аудитория	В промпте можно указать профессиональный контекст или целевую аудиторию. Это помогает выбрать терминологию, глубину
Формат результата	Указание роли не даёт модели дополнительных полномочий или гарантированной экспертности. Это прежде всего способ за

Формат результата

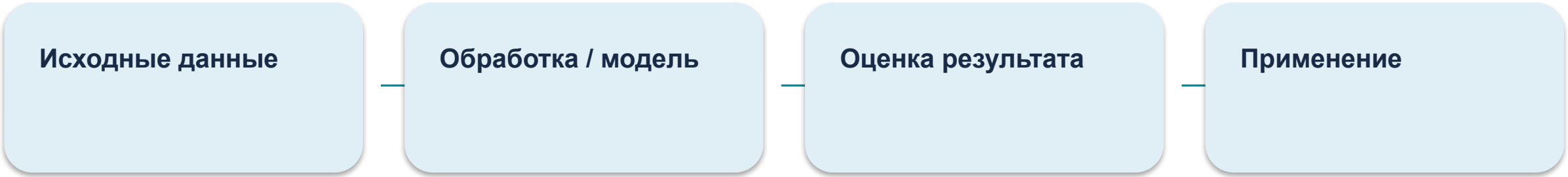
Формат лучше задавать явно, если результат должен использоваться дальше. Например: таблица с фиксированными столбцами, список из пяти пунктов, JSON-подобная структура, аннотация определённого объёма или последовательность разделов. Чем важнее машинная обработка результата, тем строже должна быть спецификация структуры.

Ключевая идея: Формат лучше задавать явно, если результат должен использоваться дальше. Например: таблица с фиксированными столбцами, список

Формат результата — продолжение

Однако после генерации структурированный ответ всё равно необходимо валидировать.

Схема: Формат результата



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Zero-Shot prompting

Zero-Shot — постановка задачи без демонстрационных примеров. Модель получает только инструкцию и входные данные. Пример: «Определи тональность каждого отзыва как положительную, нейтральную или отрицательную. Верни таблицу: №, тональность, краткое обоснование». Zero-Shot удобен, когда задача хорошо известна модели и требования легко описываются словами.

Ключевая идея: Zero-Shot — постановка задачи без демонстрационных примеров. Модель получает только инструкцию и входные данные.

Zero-Shot prompting — продолжение

Его преимущество — краткость контекста.

One-Shot и Few-Shot prompting

Рисунок 2 – Zero-Shot, One-Shot и Few-Shot prompting One-Shot включает один демонстрационный пример, Few-Shot — несколько примеров. Примеры показывают модели ожидаемое соответствие между входом и выходом.

Ключевая идея: Рисунок 2 – Zero-Shot, One-Shot и Few-Shot prompting

One-Shot и Few-Shot prompting — продолжение

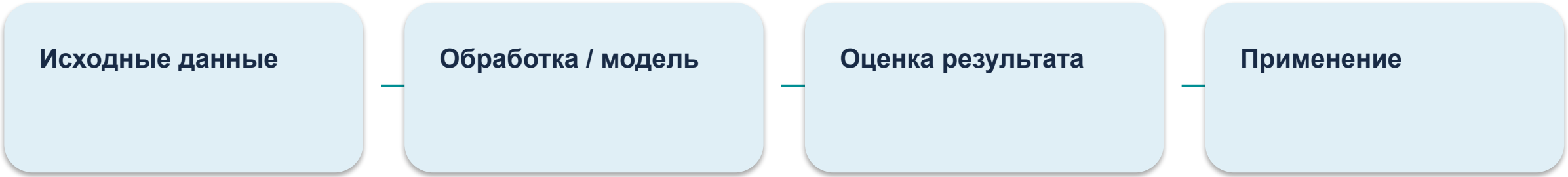
Например, при классификации обращений можно показать несколько корректно размеченных случаев: «Не могу войти в аккаунт → доступ», «Списали неизвестную сумму → платёж», после чего передать новые обращения для классификации.

Как выбирать примеры для Few-Shot

Примеры должны быть корректными, репрезентативными и согласованными. Если демонстрационные ответы противоречат друг другу, модель получает неоднозначный сигнал. Полезно включать примеры, отражающие типичные случаи и важные границы между классами. При этом не следует заполнять контекст десятками почти одинаковых примеров, если несколько качественных демонстраций уже задают правило.

Ключевая идея: Примеры должны быть корректными, репрезентативными и согласованными. Если демонстрационные ответы противоречат друг другу, мо

Схема: Как выбирать примеры для Few-Shot



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Сравнительная таблица: Как выбирать примеры для Few-Shot

Как выбирать примеры для Few-Shot	Примеры должны быть корректными, репрезентативными и согласованными. Если демонстрационные ответы противоречат друг
Структурированное рассуждение	Полезно включать примеры, отражающие типичные случаи и важные границы между классами. При этом не следует заполнять

Структурированное рассуждение

Для сложной задачи полезно декомпозировать работу на этапы: определить входные данные, выделить критерии, выполнить сравнение, проверить ограничения и только затем сформировать итоговый ответ. Такой подход уменьшает риск пропуска важных частей задачи. В учебном пособии отдельно указан Chain-of-Thought как один из методов Prompt Engineering.

Ключевая идея: Для сложной задачи полезно декомпозировать работу на этапы: определить входные данные, выделить критерии, выполнить сравнение

Структурированное рассуждение — продолжение

В практической работе безопаснее формулировать требование к проверяемым промежуточным результатам: например, попросить модель перечислить использованные критерии, показать расчёт по заданной формуле или представить краткое обоснование вывода, не требуя раскрытия внутренних скрытых рассуждений.

Декомпозиция сложной задачи

Если запрос содержит несколько независимых действий, их можно разделить на последовательные этапы. Например, анализ научной статьи можно организовать так: сначала определить цель исследования, затем извлечь данные и методы, далее выписать метрики, после этого сравнить заявленные результаты и в конце сформулировать ограничения.

Ключевая идея: Если запрос содержит несколько независимых действий, их можно разделить на последовательные этапы. Например, анализ научной с

Декомпозиция сложной задачи — продолжение

Декомпозиция особенно полезна при работе с длинными документами, программным кодом, аналитическими отчётами и задачами, где результат одного этапа используется на следующем.

Шаблон промпта для анализа документа

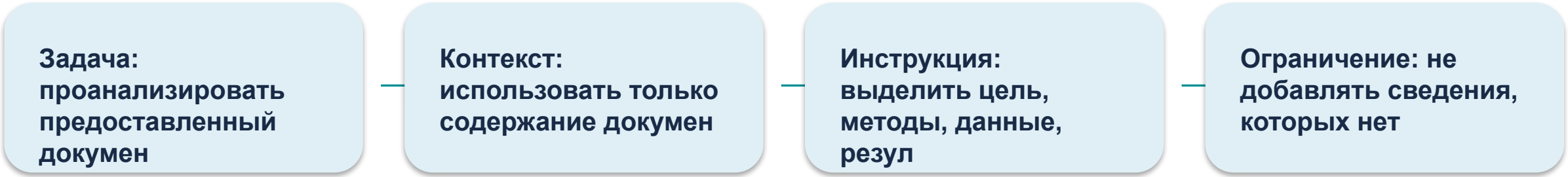
Задача: проанализировать предоставленный документ. Контекст: использовать только содержание документа. Инструкция: выделить цель, методы, данные, результаты и ограничения. Ограничение: не добавлять сведения, которых нет в документе; при отсутствии информации писать «не указано». Формат: таблица из пяти строк, затем вывод до 100 слов.

Ключевая идея: Задача: проанализировать предоставленный документ.

Шаблон промпта для анализа документа — продолжение

Проверка: числа, названия методов и выводы должны соответствовать исходному тексту. Такой шаблон не является универсальной формулой, но показывает, как соединить цель, контекст, ограничения и формат.

Схема: Шаблон промпта для анализа документа



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Шаблон промпта для сравнения моделей

Пример задания: «На основе предоставленной таблицы сравни Logistic Regression, Random Forest и нейронную сеть по accuracy, precision, recall, F1-score и времени обучения. Не пересчитывай отсутствующие показатели. Сначала сформируй сравнительную таблицу, затем укажи сильные и слабые стороны каждой модели и предложи выбор для системы, где критичен пропуск положительного класса».

Ключевая идея: Пример задания: «На основе предоставленной таблицы сравни Logistic Regression, Random Forest и нейронную сеть по accuracy, pr

Шаблон промпта для сравнения моделей — продолжение

В этом примере явно задан критерий принятия решения. Если критичен пропуск положительного класса, внимание должно быть уделено recall, а не только общей accuracy.

Сравнительная таблица: Шаблон промпта для сравнения моделей

Шаблон промпта для сравнения моделей	Пример задания: «На основе предоставленной таблицы сравни Logistic Regression, Random Forest и нейронную сеть по ac
Шаблон промпта для программирования	В этом примере явно задан критерий принятия решения. Если критичен пропуск положительного класса, внимание должно б

Шаблон промпта для программирования

При запросе программного кода следует задавать язык, версию или среду при необходимости, входные и выходные данные, ограничения и критерии проверки. Пример: «Напиши на Python функцию, которая принимает pandas DataFrame, находит числовые столбцы с пропусками и заполняет их медианой. Не изменяй категориальные столбцы. Добавь краткий пример вызова и три тестовых случая».

Ключевая идея: При запросе программного кода следует задавать язык, версию или среду при необходимости, входные и выходные данные, ограничен

Шаблон промпта для программирования — продолжение

Сгенерированный код необходимо запускать и тестировать. Корректно выглядящий код может содержать логические ошибки или небезопасные конструкции.

Итеративная разработка промпта

Рисунок 3 – Итеративный процесс совершенствования промпта Prompt Engineering — итеративный процесс. Сначала формируется базовая инструкция, затем оценивается результат, выявляются ошибки и уточняется запрос. Если модель постоянно пропускает определённый раздел, его следует явно включить в формат.

Ключевая идея: Рисунок 3 – Итеративный процесс совершенствования промпта

Итеративная разработка промпта — продолжение

Если добавляет неподтверждённые факты — усилить ограничение на источники и предусмотреть проверку. Если формат нестабилен — привести образец.

Схема: Итеративная разработка промпта



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

A/B-сравнение промптов

Два варианта промпта можно проверить на одном наборе задач и сравнить по заранее определённым критериям. Например, для извлечения реквизитов из документов измеряется доля правильно извлечённых полей, число пропусков и число добавленных моделью значений. Такой подход лучше субъективной оценки одного удачного ответа.

Ключевая идея: Два варианта промпта можно проверить на одном наборе задач и сравнить по заранее определённым критериям. Например, для извлеч

A/B-сравнение промптов — продолжение

Для учебного эксперимента можно подготовить 20–30 тестовых примеров и сравнить Zero-Shot и Few-Shot варианты.

Сравнительная таблица: A/B-сравнение промптов

A/B-сравнение промптов	Два варианта промпта можно проверить на одном наборе задач и сравнить по заранее определённым критериям. Например,
Prompt Engineering и температура генерации	Такой подход лучше субъективной оценки одного удачного ответа. Для учебного эксперимента можно подготовить 20–30 те

Prompt Engineering и температура генерации

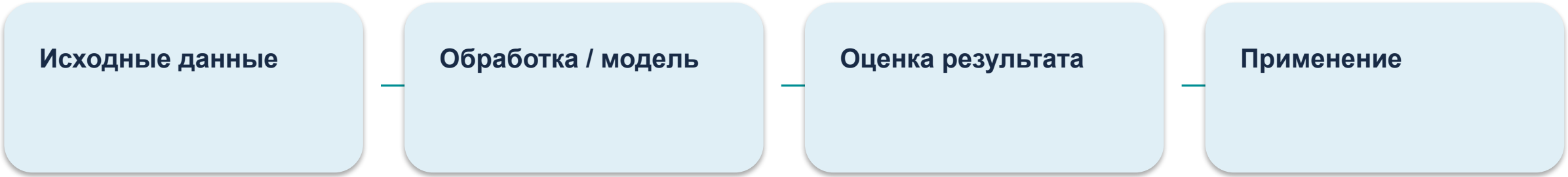
Качество результата зависит не только от текста промпта, но и от параметров генерации модели. Более высокая вариативность может быть полезна при поиске идей, а для извлечения структурированной информации обычно предпочтительнее более стабильное поведение.

Ключевая идея: Качество результата зависит не только от текста промпта, но и от параметров генерации модели. Более высокая вариативность мож

Prompt Engineering и температура генерации — продолжение

При экспериментальном сравнении промптов важно фиксировать условия запуска, иначе различия могут быть вызваны не промптом, а параметрами генерации.

Схема: Prompt Engineering и температура генерации



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Prompt Engineering и внешние документы

Если модель должна отвечать по документу, в контекст необходимо передать релевантный фрагмент и определить правила использования источника. При больших коллекциях документов предварительно выполняется поиск нужных фрагментов, после чего они включаются в контекст модели.

Ключевая идея: Если модель должна отвечать по документу, в контекст необходимо передать релевантный фрагмент и определить правила использова

Prompt Engineering и внешние документы — продолжение

В такой системе качество ответа зависит от двух частей: насколько хорошо найдена релевантная информация и насколько корректно модель использовала её при генерации.

Промпты для информационной безопасности

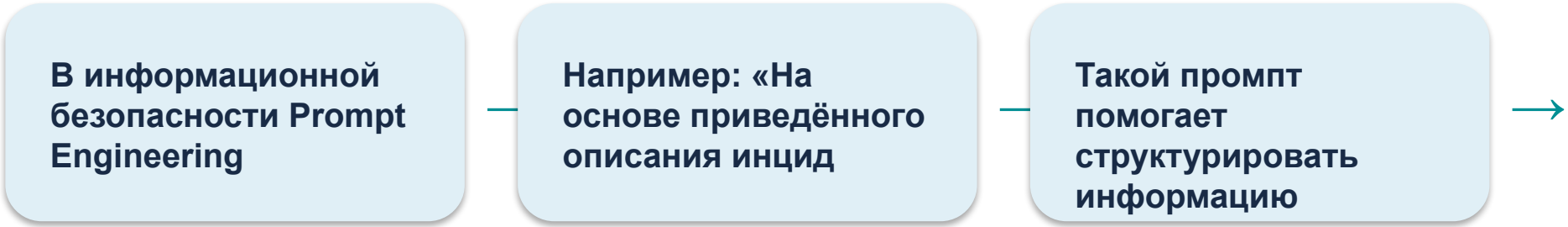
В информационной безопасности Prompt Engineering может применяться для структурирования разрешённых описаний инцидентов, классификации текстовых обращений, суммаризации отчётов и объяснения результатов аналитических инструментов. Например: «На основе приведённого описания инцидента выдели актив, событие, наблюдаемые индикаторы и неопределённые сведения.

Ключевая идея: В информационной безопасности Prompt Engineering может применяться для структурирования разрешённых описаний инцидентов, клас

Промпты для информационной безопасности — продолжение

Не придумывай отсутствующие IP-адреса, даты или технические факты. Верни результат в таблице». Такой промпт помогает структурировать информацию, но не заменяет анализ журналов и средств мониторинга.

Схема: Промпты для информационной безопасности



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Сравнительная таблица: Промпты для информационной безопасности

Промпты для информационной безопасности	В информационной безопасности Prompt Engineering может применяться для структурирования разрешённых описаний инциде
Конфиденциальные данные в промптах	Например: «На основе приведённого описания инцидента выдели актив, событие, наблюдаемые индикаторы и неопределённые
Prompt Injection	Такой промпт помогает структурировать информацию, но не заменяет анализ журналов и средств мониторинга.

Конфиденциальные данные в промптах

Промпт может содержать не только инструкцию, но и реальные данные. Поэтому перед передачей информации внешней модели необходимо проверить допустимость её обработки. Пароли, ключи доступа, персональные данные, закрытый исходный код, внутренние документы и журналы безопасности не следует передавать внешнему сервису без соответствующего разрешения, договорных и технических мер защиты.

Ключевая идея: Промпт может содержать не только инструкцию, но и реальные данные. Поэтому перед передачей информации внешней модели необходи

Prompt Injection

При использовании LLM в системах, работающих с внешними документами или веб-контентом, необходимо учитывать риск Prompt Injection. Внешний текст может содержать инструкции, пытающиеся изменить поведение модели и заставить её игнорировать исходную задачу. Защита не должна основываться только на фразе «игнорируй вредоносные инструкции».

Ключевая идея: При использовании LLM в системах, работающих с внешними документами или веб-контентом, необходимо учитывать риск Prompt Injec

Prompt Injection — продолжение

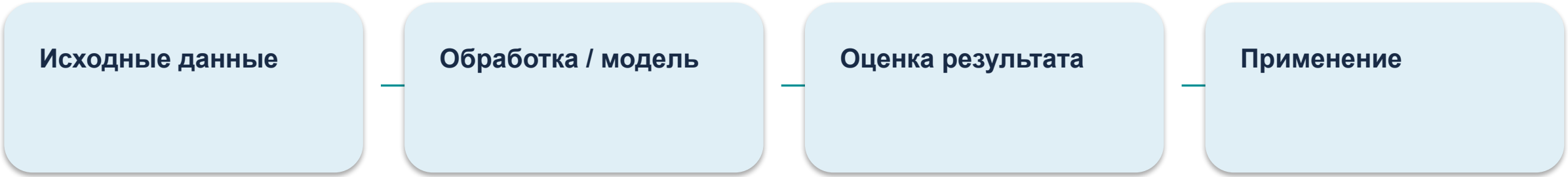
Необходимо разделять доверенные системные инструкции и недоверенные данные, ограничивать доступ модели к инструментам и данным, проверять выход и применять принцип минимальных привилегий.

Системные и пользовательские инструкции

В прикладной LLM-системе могут существовать инструкции разных уровней: правила приложения, пользовательская задача и внешние данные. Их нельзя смешивать как равнозначный текст. Например, содержимое загруженного документа должно рассматриваться как данные для анализа, а не как инструкция, автоматически изменяющая правила работы системы.

Ключевая идея: В прикладной LLM-системе могут существовать инструкции разных уровней: правила приложения, пользовательская задача и внешние

Схема: Системные и пользовательские инструкции



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Промпты и галлюцинации

Улучшение промпта может уменьшить часть ошибок, но не устраняет способность LLM генерировать неподтверждённые сведения. Поэтому запрос «не ошибайся» не является механизмом гарантии. Для фактологических задач полезно требовать опору на предоставленный источник, маркировать неизвестные сведения и проверять критичные факты после генерации.

Ключевая идея: Улучшение промпта может уменьшить часть ошибок, но не устраняет способность LLM генерировать неподтверждённые сведения. Поэто

Сравнительная таблица: Промпты и галлюцинации

Промпты и галлюцинации	Улучшение промпта может уменьшить часть ошибок, но не устраняет способность LLM генерировать неподтверждённые сведе
Prompt Engineering в научной работе	Для фактологических задач полезно требовать опору на предоставленный источник, маркировать неизвестные сведения и п

Prompt Engineering в научной работе

В научной работе промпты могут использоваться для структурирования собственных материалов, редактирования формулировок, подготовки вариантов классификации или помощи с программным кодом. При этом модель не должна быть источником вымышленных публикаций, экспериментальных данных или неподтверждённых выводов.

Ключевая идея: В научной работе промпты могут использоваться для структурирования собственных материалов, редактирования формулировок, подго

Prompt Engineering в научной работе — продолжение

Библиографические записи необходимо проверять по реальным источникам и оформлять по установленному стандарту.

Prompt Engineering в образовании

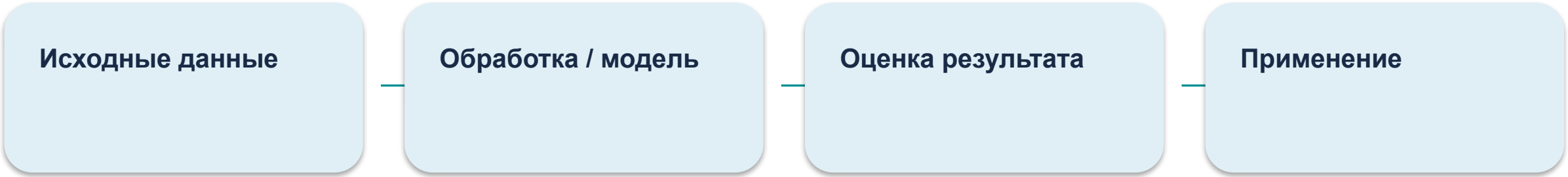
В образовательной задаче полезны промпты, которые не просто выдают готовый ответ, а организуют объяснение, сравнение и самопроверку. Например: «Объясни различие precision и recall, затем предложи две небольшие задачи без ответов, после моего решения проверь вычисления».

Ключевая идея: В образовательной задаче полезны промпты, которые не просто выдают готовый ответ, а организуют объяснение, сравнение и самопр

Prompt Engineering в образовании — продолжение

Для преподавателя Prompt Engineering может использоваться при подготовке черновиков вопросов, кейсов и примеров, но содержание должно быть проверено перед включением в учебно-методические материалы.

Схема: Prompt Engineering в образовании



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Практический кейс: Zero-Shot и Few-Shot

Задача — классифицировать обращения пользователей по категориям «доступ», «платёж», «техническая ошибка». В Zero-Shot варианте модель получает определения трёх классов и новые обращения. В Few-Shot варианте дополнительно приводятся несколько корректно размеченных примеров. Эксперимент следует проводить на одинаковом тестовом наборе.

Ключевая идея: Задача — классифицировать обращения пользователей по категориям «доступ», «платёж», «техническая ошибка». В Zero-Shot вариант

Практический кейс: Zero-Shot и Few-Shot — продолжение

Затем сравнивается число правильных классификаций и анализируются ошибки. Это позволяет эмпирически определить, дают ли примеры преимущество для конкретной задачи.

Практический кейс: извлечение информации

Пусть требуется извлечь из текста документа номер, дату, организацию и сумму. Промпт должен перечислять поля, определить поведение при отсутствии значения и запретить угадывание. Пример требования: «Если поле отсутствует, верни null. Не восстанавливай значение по контексту и не используй внешние знания». После генерации результат сравнивается с эталонной разметкой.

Ключевая идея: Пусть требуется извлечь из текста документа номер, дату, организацию и сумму. Промпт должен перечислять поля, определить пове

Сравнительная таблица: Практический кейс: извлечение информации

Практический кейс: извлечение информации	Пусть требуется извлечь из текста документа номер, дату, организацию и сумму. Промпт должен перечислять поля, опред
Практический кейс: сравнение двух ответов	Пример требования: «Если поле отсутствует, верни null. Не восстанавливай значение по контексту и не используй внешн

Практический кейс: сравнение двух ответов

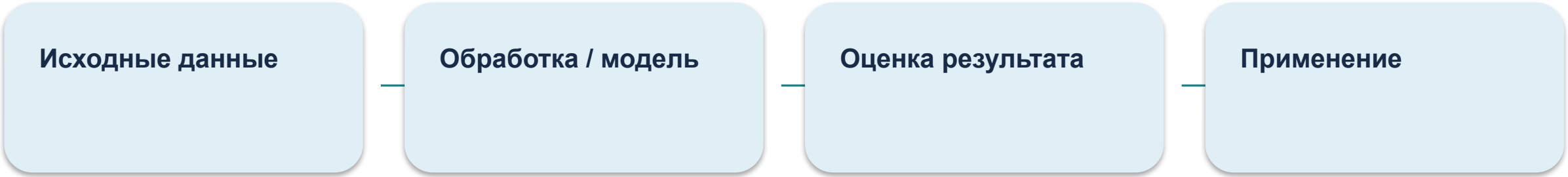
Магистрантам можно дать один и тот же исходный текст и два промпта: короткий общий и структурированный. После получения ответов они оценивают полноту, точность, соблюдение формата и количество неподтверждённых утверждений.

Ключевая идея: Магистрантам можно дать один и тот же исходный текст и два промпта: короткий общий и структурированный. После получения ответ

Практический кейс: сравнение двух ответов — продолжение

Такое упражнение показывает, что Prompt Engineering следует рассматривать не как набор «магических фраз», а как инженерный процесс постановки задачи и экспериментальной проверки.

Схема: Практический кейс: сравнение двух ответов



Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Рекомендуемая последовательность проектирования промпта

Определить конечную задачу и критерий успешного результата. Выделить минимально необходимый контекст. Сформулировать однозначную инструкцию. Указать ограничения и допустимые источники. Задать формат ответа. Решить, нужны ли Zero-Shot или Few-Shot примеры. Проверить промпт на наборе тестовых случаев. Проанализировать ошибки и уточнить инструкцию.

Ключевая идея: Определить конечную задачу и критерий успешного результата.

Рекомендуемая последовательность проектирования промпта — продолжение

Проверить безопасность данных и итогового результата.

Основные выводы

Prompt Engineering — инженерный процесс проектирования, тестирования и улучшения запросов к моделям ИИ. Эффективный промпт обычно определяет цель, контекст, инструкцию, ограничения и формат результата. Zero-Shot работает без примеров, One-Shot использует один пример, Few-Shot — несколько демонстраций. Сложные задачи полезно декомпозировать на проверяемые этапы.

Ключевая идея: Prompt Engineering — инженерный процесс проектирования, тестирования и улучшения запросов к моделям ИИ.

Основные выводы — продолжение

Качество промпта следует оценивать на наборе тестовых примеров, а не по одному удачному ответу. Prompt Engineering не устраняет галлюцинации и не заменяет проверку фактов. При работе с промптами необходимо учитывать конфиденциальность, Prompt Injection и разделение доверенных инструкций и внешних данных.

Контрольные вопросы

1. Что такое Prompt Engineering и почему формулировка запроса влияет на результат LLM? 2. Какие компоненты целесообразно включать в структурированный промпт? 3. Чем отличаются Zero-Shot, One-Shot и Few-Shot prompting и в каких случаях они применяются? 4. Как следует подбирать демонстрационные примеры для Few-Shot prompting? 5.

Ключевая идея: 1. Что такое Prompt Engineering и почему формулировка запроса влияет на результат LLM?

Контрольные вопросы — продолжение

Как декомпозиция задачи и структурированные промежуточные результаты помогают при решении сложных задач? 6. Какие критерии можно использовать для экспериментальной оценки качества промпта? 7. Какие типичные ошибки возникают при проектировании промптов и как их исправлять? 8. Как Prompt Engineering применяется при анализе документов, программировании и задачах информационной безопасности?

Контрольные вопросы — продолжение

9. Какие риски связаны с конфиденциальными данными и Prompt Injection? 10. Почему даже хорошо спроектированный промпт не отменяет необходимость проверки фактов и результатов модели?

Схема: Контрольные вопросы

1. Что такое Prompt Engineering и почему формули
2. Какие компоненты целесообразно включать в стр
3. Чем отличаются Zero-Shot, One-Shot и Few-Shot
4. Как следует подбирать демонстрационные пример

Последовательность отражает основные этапы рассматриваемого процесса; конкретная реализация зависит от задачи.

Сравнительная таблица: Контрольные вопросы

Контрольные вопросы	1. Что такое Prompt Engineering и почему формулировка запроса влияет на результат LLM?
Список источников	2. Какие компоненты целесообразно включать в структурированный промпт?

Список источников

1. Бессмертный, И. А. Системы искусственного интеллекта : методическое пособие / И. А. Бессмертный, А. Е. Авдюшина, А. В. Кугаевских, Ю. А. Королева, Н. Г. Рущенко. – Санкт-Петербург : Университет ИТМО, 2024. – 281 с. 2. Соснило, А. И. Атлас искусственного интеллекта для бизнеса и власти : учебное пособие / А. И. Соснило. – Санкт-Петербург : Университет ИТМО, 2022. – 98 с. 3.

Ключевая идея: 1. Бессмертный, И. А. Системы искусственного интеллекта : методическое пособие / И. А. Бессмертный, А. Е. Авдюшина, А. В. Куг

Список источников — продолжение

Сидоркина, И. Г. Системы искусственного интеллекта : учебное пособие / И. Г. Сидоркина. – Москва : КНОРУС, 2018. – 248 с. 4. Баланов, А. Н. Искусственный интеллект. Понимание, применение и перспективы : учебник для вузов / А. Н. Баланов. – Санкт-Петербург : Лань, 2024. – 312 с. 5. Калинин, М. О. Основы искусственного интеллекта.

Список источников — продолжение

Безопасность искусственного интеллекта : учебное пособие / М. О. Калинин, В. М. Крундышев. – Санкт-Петербург, 2024. – 220 с. 6. Загорулько, Ю. А. Искусственный интеллект. Инженерия знаний : учебное пособие для вузов / Ю. А. Загорулько, Г. Б. Загорулько. – Москва : Юрайт, 2019. – 184 с. 7. Андреянов, Н. В. Методы искусственного интеллекта : учебное пособие / Н. В. Андреянов, Т. С.

Список источников — продолжение

Евдокимова, А. Д. Павлов [и др.]. – Казань : КНИТУ-КАИ, 2024. – 392 с. 8. Основы искусственного интеллекта : учебное пособие. – Гл. 13 : Prompt Engineering: принципы создания промптов, Zero-Shot, Few-Shot, Chain-of-Thought, практические кейсы.

График: иллюстрация зависимости



Иллюстративные значения (не результаты эксперимента).

Итоги и вопросы для обсуждения

1. Объясните понятие «Понятие Prompt Engineering» и приведите пример.

2. Объясните понятие «Почему формулировка запроса влияет на результат» и приведите пример.

3. Объясните понятие «Основные компоненты эффективного промпта» и приведите пример.

4. Объясните понятие «Конкретность и однозначность» и приведите пример.

5. Объясните понятие «Контекст в промпте» и приведите пример.

6. Объясните понятие «Роль и аудитория» и приведите пример.

7. Объясните понятие «Формат результата» и приведите пример.