

1. Теоретические основы многоэтапности в промпт-инжиниринге

Многоэтапность (Multi-step reasoning) — это метод улучшения взаимодействия с языковыми моделями, при котором сложные задачи разбиваются на последовательные, логически связанные этапы. Этот подход имеет прочное научное обоснование в когнитивной науке и теории вычислений.

Ключевые исследования, лежащие в основе многоэтапности

1. Chain-of-Thought (CoT) — Цепочка размышлений

Фундаментальное исследование, которое впервые формализовало подход "думай шаг за шагом" для языковых моделей. CoT позволяет модели явно демонстрировать ход рассуждений, что приводит к значительному улучшению результатов в задачах, требующих сложных рассуждений.

2. Процедурные знания в предварительном обучении

Исследование доказывает, что языковые модели опираются на процедурные знания, полученные во время предобучения. Это научное обоснование того, почему разбиение сложных задач на последовательность простых шагов улучшает результаты.

3. Strategic Chain-of-Thought

Исследование, демонстрирующее преимущество предварительного планирования стратегии решения задачи перед выполнением самих шагов.

4. LADDER: Рекурсивная декомпозиция задач

Исследование о самоулучшающихся языковых моделях через рекурсивную декомпозицию задач. Показывает, как модели могут эффективно решать сложные задачи, разбивая их на более простые подзадачи и последовательно их решая.

5. InftyThink: Преодоление ограничений длины контекста

Исследование, посвященное методике разбиения длинных рассуждений на сегменты с промежуточными выводами для преодоления ограничений контекстного окна.

2. Почему многоэтапность работает: научное обоснование

Когнитивная база

1. **Снижение когнитивной нагрузки:** Разбиение сложных задач на этапы снижает когнитивную нагрузку на модель, позволяя ей сфокусироваться на решении каждой подзадачи по отдельности.
2. **Иерархическая обработка информации:** Исследования показывают, что языковые модели лучше обрабатывают информацию, когда она представлена в иерархической структуре (от общего к частному или наоборот).
3. **Активация процедурных знаний:** Многоэтапность активирует процедурные знания, встроенные в модель во время предобучения, что позволяет применять более специализированные алгоритмы решения.

Технические основания

1. **Структурированное пространство решений:** Поэтапное решение сужает пространство поиска решений на каждом этапе, снижая вероятность ошибок.
2. **Смягчение эффекта затухания градиентов:** В длинных последовательностях токенов ранние этапы рассуждений могут "забываться". Структурирование рассуждений решает эту проблему.
3. **Улучшение трассируемости рассуждений:** Явные промежуточные шаги позволяют отследить ход мысли и выявить источники ошибок.

3. Основные принципы применения многоэтапности

Принцип 1: Декомпозиция сложных задач

Суть: Разделение сложной задачи на автономные, но связанные подзадачи.

Как применять:

- Анализируйте задачу для выявления логически отделимых компонентов
- Определяйте зависимости между подзадачами
- Создавайте четкую последовательность выполнения

Пример из исследований:

Задача решения математической проблемы

1. Сначала проанализируй условия задачи и выдели ключевые данные
2. Определи, какие математические концепции необходимо применить
3. Составь план решения с четкими этапами
4. Реши каждый этап, проверяя промежуточные результаты
5. Проведи финальную проверку всего решения

Принцип 2: Иерархическое структурирование

Суть: Организация процесса решения в иерархическую структуру, от высокоуровневого плана к детальным шагам.

Как применять:

- Начинайте с макроуровня (общей стратегии)
- Постепенно углубляйтесь в детали каждого компонента
- Обеспечивайте связность между уровнями

Пример из исследований:

Структура анализа текста

Уровень 1: Общий план анализа

- Определить тему и основной тезис
- Выявить ключевые аргументы
- Оценить использованные источники и доказательства

Уровень 2: Детальный анализ каждого компонента

Анализ основного тезиса

- Определить явный и скрытый посыл
- Оценить четкость формулировки
- Проверить соответствие заявленной теме

Анализ аргументов

...

Принцип 3: Промежуточная верификация

Суть: Проверка корректности результата каждого этапа перед переходом к следующему.

Как применять:

- Определяйте критерии успешного выполнения каждого этапа
- Включайте шаги самопроверки

- Документируйте промежуточные результаты

Пример из исследования:

Пример промежуточной верификации в математике

1. Вычисли произведение $345 \times 7 = ?$
2. Проверка: проверим результат используя разбиение: $345 \times 7 = 300 \times 7 + 45 \times 7 = 2100 + 315 = 2415$
3. Теперь вычисли $345 \times 20 = ?$
4. Проверка: $345 \times 20 = 345 \times 2 \times 10 = 690 \times 10 = 6900$
5. Сложи полученные результаты: $2415 + 6900 = ?$
6. Проверка: $2415 + 6900 = 2400 + 6900 + 15 = 9300 + 15 = 9315$
7. Итоговый ответ: $345 \times 27 = 9315$

Принцип 4: Адаптивная детализация

Суть: Изменение уровня детализации шагов в зависимости от сложности задачи.

Как применять:

- Для сложных частей увеличивайте количество промежуточных шагов
- Для простых участков можно использовать более крупные шаги
- Учитывайте контекст и предметную область задачи

Пример из исследования "Chain of Draft":

Подход с переменной детализацией

Задача: Вычислить производную функции $f(x) = \sin(x^2) \times \log(x+3)$

1. Сначала применим правило произведения [высокий уровень детализации]

- Определить $f(x) = u(x) \times v(x)$, где $u(x) = \sin(x^2)$ и $v(x) = \log(x+3)$

- Найти $u'(x)$:

* Применить правило цепи

* $g(x) = x^2$, $g'(x) = 2x$

* $u'(x) = \cos(x^2) \times 2x$

- Найти $v'(x)$:

* $v'(x) = 1/(x+3)$

2. Объединить результаты [низкий уровень детализации]

- $f'(x) = u'(x)v(x) + u(x)v'(x)$

- $f'(x) = \cos(x^2) \times 2x \times \log(x+3) + \sin(x^2) \times 1/(x+3)$

Принцип 5: Итеративное улучшение

Суть: Последовательное улучшение решения через оценку и корректировку.

Как применять:

- После получения первичного решения, анализируйте его на предмет ошибок
- Улучшайте отдельные этапы или аспекты решения
- Повторяйте процесс до достижения необходимого качества

Пример из исследования "Self-Refine":

```
# Задача написания кода
## Итерация 1: Создание базового решения
[Базовое решение функции]

## Итерация 2: Анализ недостатков
- Производительность можно улучшить
- Не обрабатываются некоторые крайние случаи
- Код не достаточно читаемый

## Итерация 3: Улучшенная версия
[Улучшенный код с комментариями и оптимизацией]
```

4. Варианты и модификации многоэтапного подхода

Chain-of-Thought (CoT)

Базовый подход, при котором модель явно демонстрирует ход размышлений. Эффективен для большинства задач рассуждения.

Пример (из исследования):

Задача: В магазине продаются футболки по 800 рублей и джинсы по 2400 рублей. Маша купила 3 футболки и 2 пары джинсов. Сколько всего денег она потратила?

Решение (CoT):

Сначала посчитаем стоимость футболок:
3 футболки × 800 рублей = 2400 рублей.

Теперь посчитаем стоимость джинсов:

2 пары джинсов × 2400 рублей = 4800 рублей.

Общая стоимость:

2400 рублей + 4800 рублей = 7200 рублей.

Ответ: Маша потратила 7200 рублей.

Chain of Draft (CoD)

Облегченная версия CoT, где промежуточные рассуждения даются в краткой, телеграфной форме. Эффективен для снижения количества токенов при сохранении качества рассуждений.

Пример (из исследования):

Задача: [та же задача про покупки Маши]

Решение (CoD):

- Футболки: $3 \times 800 = 2400$

- Джинсы: $2 \times 2400 = 4800$

- Всего: $2400 + 4800 = 7200$

- Ответ: 7200 рублей

Program of Thought (PoT)

Использование программного кода или псевдокода для структурирования решения. Особенно эффективен для математических и алгоритмических задач.

Пример (из исследования):

Задача: [та же задача про покупки Маши]

Решение (PoT):

```
```python
```

```
Объявляем переменные
```

```
price_tshirt = 800 # цена футболки в рублях
```

```
price_jeans = 2400 # цена джинсов в рублях
```

```
quantity_tshirt = 3 # количество купленных футболок
```

```
quantity_jeans = 2 # количество купленных джинсов
```

```
Вычисляем стоимость по категориям
```

```
cost_tshirt = price_tshirt * quantity_tshirt
```

```
cost_jeans = price_jeans * quantity_jeans
```

```
Вычисляем общую стоимость
total_cost = cost_tshirt + cost_jeans

print(f"Маша потратила {total_cost} рублей")
```

Общая стоимость: 7200 рублей

### ### Tree of Thoughts (ToT)

Рассмотрение нескольких альтернативных путей рассуждения с оценкой и выбором наиболее перспективных. Эффективен для задач с неоднозначными решениями.

**Пример** (из исследования ReasonGraph):

Задача: Найти оптимальный маршрут из точки A в точку E через промежуточные пункты.

Корневая мысль: Нужно найти маршрут  $A \rightarrow E$  с минимальной стоимостью.

Ветвь 1: Маршрут через B  $\hookleftarrow A \rightarrow B$  стоит 5  $\hookleftarrow B \rightarrow E$  стоит 8  $\hookleftarrow$  Общая стоимость: 13

Ветвь 2: Маршрут через C  $\hookleftarrow A \rightarrow C$  стоит 3  $\hookleftarrow C \rightarrow E$  стоит 7  $\hookleftarrow$  Общая стоимость: 10

Ветвь 3: Маршрут через D  $\hookleftarrow A \rightarrow D$  стоит 4  $\hookleftarrow D \rightarrow E$  стоит 4  $\hookleftarrow$  Общая стоимость: 8

Оценка: Маршрут через D имеет наименьшую стоимость (8) Ответ:  
Оптимальный маршрут  $A \rightarrow D \rightarrow E$

### ### Chain-of-Verification (CoV)

Дополнение к базовым подходам, добавляющее этап критической проверки полученного решения. Эффективен для уменьшения ошибок.

**Пример** (из исследования о проверке математических ошибок):

Задача: Вычислить значение выражения  $(15 \times 8) - (12 \div 4) + 7$

Решение:

1.  $(15 \times 8) = 120$
2.  $(12 \div 4) = 3$
3.  $120 - 3 + 7 = 124$

Верификация:

- Проверим шаг 1:  $15 \times 8 = 15 \times (2^3) = 15 \times 4 \times 2 = 60 \times 2 = 120 \checkmark$
- Проверим шаг 2:  $12 \div 4 = 3 \checkmark$
- Проверим шаг 3:  $120 - 3 = 117$ , затем  $117 + 7 = 124 \checkmark$
- Проверим общий порядок операций: согласно правилам, умножение/деление выполняются перед сложением/вычитанием  $\checkmark$

Ответ: 124

### ### LADDER (Recursive Task Decomposition)

Рекурсивный подход к декомпозиции задач, где сложные задачи сначала разбиваются на упрощенные версии, решаются, а затем знания переносятся на исходную задачу.

**\*\*Пример\*\*** (из исследования):

Задача: Решить интеграл  $\int (x^2 + 3x + 2)/(x^2 - 4) dx$

1. Упрощенная задача 1: Найти интеграл  $\int (1)/(x-2) dx$  Решение:  $\ln|x-2| + C$
2. Упрощенная задача 2: Найти интеграл  $\int (1)/(x+2) dx$  Решение:  $\ln|x+2| + C$
3. Теперь решим основную задачу через разложение на простые дроби:  $(x^2 + 3x + 2)/(x^2 - 4) = (x^2 + 3x + 2)/((x-2)(x+2))$
4. Применим метод неопределенных коэффициентов:  $(x^2 + 3x + 2)/((x-2)(x+2)) = A/(x-2) + B/(x+2)$
5. [Дальнейшие математические выкладки]
6. Получаем  $A = 3/2$ ,  $B = -1/2$
7. Итоговый интеграл:  $\int (x^2 + 3x + 2)/(x^2 - 4) dx = (3/2)\int (1)/(x-2) dx - (1/2)\int (1)/(x+2) dx$
8. Используя решения упрощенных задач:  $(3/2)\ln|x-2| - (1/2)\ln|x+2| + C$

### InftyThink



Методология для преодоления ограничений контекстного окна путем разбиения длинных рассуждений на сегменты с переносом информации между ними через промежуточные резюме.

**\*\*Пример\*\*** (из исследования):

Задача: Доказать теорему [сложная математическая задача]

Сегмент 1: [500-1000 слов рассуждений]

Резюме Сегмента 1:

- Установлены базовые условия теоремы
- Доказано промежуточное утверждение A
- Следующие шаги: доказать утверждения B и C

Сегмент 2 (опирается на резюме Сегмента 1): [Продолжение рассуждений на основе предыдущего резюме]

Резюме Сегмента 2:

- Доказано утверждение B
- Установлена связь между утверждениями A и B
- Следующие шаги: доказать утверждение C и обобщить результат

[Продолжение сегментов...]

Финальное заключение

## 5. Практические примеры применения многоэтапного подхода

### Пример 1: Анализ данных с Chain-of-Thought

## Задача анализа финансовых данных

### Этап 1: Предварительный анализ данных

- Определи основные показатели: среднее значение, медиану, размах
- Проанализируй тренды и сезонность в данных

- Выяви выбросы и аномалии

## Этап 2: Сравнительный анализ

- Сравни текущие показатели с историческими данными
- Сравни показатели с отраслевыми бенчмарками
- Определи ключевые отклонения и их значимость

## Этап 3: Анализ причинно-следственных связей

- Определи факторы, влияющие на ключевые показатели
- Оцени силу влияния каждого фактора
- Выяви взаимозависимости между факторами

## Этап 4: Формирование рекомендаций

- Сформулируй практические выводы из анализа
- Предложи конкретные действия для улучшения показателей
- Укажи метрики для отслеживания результатов

## Этап 5: Оценка рисков

- Определи потенциальные риски предложенных действий
- Предложи меры по минимизации каждого риска
- Укажи триггеры для пересмотра стратегии

### Пример 2: Разработка программы с Program of Thought (PoT)

# Задача: Написать функцию для вычисления факториала числа с проверкой на ошибки

## Этап 1: Определение требований и план

- Функция должна принимать целое число  $n$
- Должна возвращать  $n!$  для  $n \geq 0$
- Должна корректно обрабатывать ошибки ввода

## Этап 2: Разработка алгоритма в псевдокоде

```
Псевдокод функции factorial:
function factorial(n)
 if n не является целым числом:
 вернуть ошибку "Ввод должен быть целым числом"

 if n < 0:
 вернуть ошибку "Ввод должен быть неотрицательным"

 if n == 0:
 вернуть 1

 result = 1
 for i от 1 до n:
 result = result * i

 вернуть result
```

## Этап 3: Реализация в Python

```
def factorial(n):
 # Проверка типа ввода
 if not isinstance(n, int):
 raise TypeError("Ввод должен быть целым числом")

 # Проверка на отрицательное значение
 if n < 0:
 raise ValueError("Ввод должен быть неотрицательным")

 # Базовый случай
 if n == 0:
 return 1

 # Вычисление факториала
 result = 1
 for i in range(1, n + 1):
 result *= i

 return result
```

## Этап 4: Тестирование

- Тест 1: factorial(5) должен вернуть 120
- Тест 2: factorial(0) должен вернуть 1
- Тест 3: factorial(-1) должен вызвать ValueError
- Тест 4: factorial("string") должен вызвать TypeError

### Пример 3: Решение проблемы с Tree of Thoughts (ToT)

## Задача: Определить оптимальную стратегию выхода на новый рынок

### Корневая проблема:

Какую стратегию выбрать для выхода компании на новый рынок?

#### Ветвь 1: Самостоятельный вход

├─ Преимущества: | ├─ Полный контроль над операциями | ├─ 100% прибыли остается в компании | └─ Свобода в бизнес-решениях | ├─ Недостатки: | ├─ Высокие начальные инвестиции (оценка: \$2-3М) | ├─ Длительный период окупаемости (18-24 месяца) | ├─ Отсутствие локальной экспертизы | └─ Высокие регуляторные риски | └─ Оценка жизнеспособности: Средняя (6/10)

#### Ветвь 2: Совместное предприятие с локальным игроком

├─ Преимущества: | ├─ Доступ к существующей инфраструктуре | ├─ Локальная экспертиза и связи | ├─ Разделение рисков | └─ Ускоренный вход на рынок (6-8 месяцев) | ├─ Недостатки: | ├─ Разделение прибыли (40-60%) | ├─ Потенциальные конфликты в управлении | └─ Ограниченный контроль | └─ Оценка жизнеспособности: Высокая (8/10)

#### Ветвь 3: Лицензирование технологии/продукта

├─ Преимущества: | ├─ Минимальные инвестиции | ├─ Быстрый выход на рынок (3-4 месяца) | ├─ Нулевые операционные расходы | └─ Минимальные риски | ├─ Недостатки: | ├─ Низкий доход (роялти 5-15%) | ├─ Отсутствие контроля над маркетингом | └─ Риск ущерба репутации при плохой

реализации | — Создание потенциального конкурента | — Оценка  
жизнеспособности: Средняя (5/10)

## Анализ и сравнение:

Учитывая текущие финансовые возможности компании (\$1.5M для инвестиций), приоритет быстрого входа на рынок и важность локальной экспертизы, наиболее оптимальным вариантом представляется Ветвь 2 (Совместное предприятие).

## Рекомендуемые следующие шаги:

1. Идентифицировать 3-5 потенциальных партнеров
2. Разработать критерии оценки партнеров
3. Подготовить финансовую модель СП
4. Определить структуру управления и разделения обязанностей

### Пример 4: Юридический анализ с Chain-of-Verification (CoV)

# Задача: Проанализировать контракт на предмет юридических рисков

## Этап 1: Первичный анализ

- Определить предмет контракта: Поставка программного обеспечения и услуг по внедрению
- Проанализировать ключевые условия: сроки, цена, условия оплаты, порядок приемки
- Идентифицировать стороны и их обязательства

## Этап 2: Анализ рисков

- Риск 1: Нечеткие критерии приемки работ
  - Пункт 4.3 не определяет конкретные параметры успешной приемки
  - Отсутствует процедура согласования тестовых сценариев

- Риск 2: Неограниченная ответственность поставщика
  - Пункт 7.2 не устанавливает предел ответственности
  - Потенциально может превысить стоимость контракта
- Риск 3: Интеллектуальные права на модификации
  - Нет четкого разграничения прав на модификации ПО

## Этап 3: Верификация анализа

- Перепроверка риска 1:
  - Повторное изучение разделов 4.1-4.7
  - Сравнение с отраслевыми стандартами
  - Проверка сопутствующих документов (ТЗ, SLA)
  - Результат: риск подтверждается
- Перепроверка риска 2:
  - Анализ раздела 7 "Ответственность сторон"
  - Проверка наличия не прямых ограничений ответственности
  - Сопоставление с законодательными ограничениями
  - Результат: риск подтверждается частично (есть некоторые не прямые ограничения)
- Перепроверка риска 3:
  - Детальный анализ раздела 9 "Интеллектуальная собственность"
  - Проверка возможных противоречий между пунктами
  - Оценка применимого законодательства
  - Результат: риск подтверждается

## Этап 4: Рекомендации

1. Для риска 1: Добавить приложение с детальными критериями приемки
2. Для риска 2: Установить лимит ответственности в размере 100% стоимости контракта
3. Для риска 3: Детализировать права на модификации ПО

## 6. Комбинация многоэтапности с другими методами

### Многоэтапность + Специфичность

Комбинирование пошагового подхода с высокоспецифичными инструкциями для

каждого этапа.

**\*\*Пример\*\***:

## Задача исследования рынка

### Этап 1: Анализ конкурентов (с высокой специфичностью)

- Исследуй ТОП-5 конкурентов в сегменте премиальных электромобилей
- Для каждого конкурента проанализируй:
  - Ценовую политику (базовая цена, опции, стоимость обслуживания)
  - Технические характеристики (запас хода, мощность, время зарядки)
  - Стратегии позиционирования (экологичность/luxury/технологичность)
- Создай сравнительную таблицу с количественными показателями
- Определи ключевые отличия и уникальные преимущества каждого конкурента

### Этап 2: Анализ потребителей...

### Многоэтапность + Ролевая специализация

Назначение разных ролей для разных этапов процесса.

**\*\*Пример\*\***:

## Процесс разработки продукта

### Этап 1: Генерация идей (Роль: Креативный дизайнер)

Как опытный креативный дизайнер с 15-летним опытом в потребительской электронике, предложи 10 инновационных концепций для переосмысления традиционного пульта дистанционного управления для умного дома. Фокусируйся на нестандартных решениях, выходящих за рамки привычных форм-факторов.

## Этап 2: Практическая оценка (Роль: Инженер-разработчик)

Как инженер-разработчик с опытом в производстве потребительской электроники, оцени техническую реализуемость каждой из предложенных концепций. Оцени сложность производства, предполагаемую себестоимость, технологические ограничения.

## Этап 3: Маркетинговая оценка...

### Многоэтапность + Self-consistency

Многократное решение одной задачи с последующим сравнением результатов.

\*\*Пример\*\*:

## Задача решения сложной инвестиционной проблемы

### Подход 1: Дисконтирование денежных потоков

[Пошаговое решение с применением метода DCF] Результат: Рекомендуемая цена акции \$45-50

### Подход 2: Мультипликаторы

[Пошаговое решение с применением сравнительного анализа P/E, EV/EBITDA] Результат: Рекомендуемая цена акции \$48-53

### Подход 3: Сценарный анализ

[Пошаговое решение с моделированием оптимистичного, пессимистичного и базового сценариев] Результат: Рекомендуемая цена акции \$43-55

## Консолидированный анализ

- Все три подхода показывают пересечение в диапазоне \$48-50



- Наибольшую уверенность дает метод мультипликаторов из-за наличия близких аналогов
- Рекомендуемая целевая цена: \$49 с диапазоном \$48-50

## ## 7. Лучшие практики и рекомендации

### ### Когда применять многоэтапный подход

- **Сложные аналитические задачи**, требующие нескольких логических шагов
- **Задачи с высоким риском ошибок**, где важна промежуточная проверка
- **Творческие задачи**, требующие структурированного подхода
- **Обучающие задачи**, где важно продемонстрировать процесс мышления

### ### Оптимизация многоэтапных промптов

1. **Балансируйте детализацию и краткость**: используйте подход Chain of Draft для снижения количества токенов без потери качества рассуждений
2. **Указывайте целью каждого этапа**: помогает модели понять, к чему она должна прийти
3. **Включайте механизмы коррекции**: добавляйте инструкции по обнаружению и исправлению ошибок
4. **Адаптируйте к конкретной задаче**: выбирайте подходящий вариант многоэтапности (CoT, ToT, PoT) в зависимости от типа задачи
5. **Используйте визуальное форматирование**: разделяйте этапы заголовками, списками и маркерами для улучшения понимания

## ## 8. Ограничения и предостережения

1. **Увеличение количества токенов**: многоэтапные промпты обычно значительно длиннее, что может увеличивать стоимость запросов
2. **Сложность составления**: разработка эффективных многоэтапных промптов требует времени и экспертизы
3. **Потенциальные ловушки рассуждения**: иногда модель может "застрять" в неправильной логике и пронести ошибку через все этапы
4. **Не всегда необходимо**: для простых задач многоэтапный подход может быть

избыточным

## ## Заключение

Многоэтапность — один из наиболее мощных и научно-обоснованных методов промпт-инжиниринга. Исследования убедительно доказывают, что разбиение сложных задач на последовательные, структурированные шаги значительно улучшает способность языковых моделей решать сложные задачи.

Ключ к эффективному применению многоэтапности — выбор подходящего варианта метода (CoT, ToT, PoT и др.) в зависимости от специфики задачи и умелое комбинирование с другими техниками промпт-инжиниринга.

Практика показывает, что даже относительно простое добавление инструкции "думай шаг за шагом" может значительно улучшить результаты, а более сложные структурированные подходы способны раскрыть полный потенциал языковых моделей в решении самых сложных задач.