

# Масштабируемый выбор лучших из N для больших языковых моделей с помощью самоуверенности

Дата: 2025-02-25 00:00:00

Ссылка на исследование: <https://arxiv.org/pdf/2502.18581>

Рейтинг: 75

Адаптивность: 85

## Ключевые выводы:

Исследование представляет новый метод 'Self-certainty' для улучшения процесса выбора лучшего ответа из нескольких вариантов (Best-of-N selection) в больших языковых моделях (LLM). Основная цель - создать эффективный метрик для оценки качества ответов без использования внешних моделей вознаграждения. Результаты показывают, что Self-certainty эффективно масштабируется с увеличением количества образцов, улучшает рассуждения в цепочке мыслей (Chain-of-Thought) и обобщается на задачи с открытыми ответами.

## Объяснение метода:

Self-Certainty предлагает эффективный способ оценки уверенности LLM в ответах без внешних моделей. Метод позволяет выбирать лучшие ответы из нескольких вариантов, работает с открытыми задачами и масштабируется с увеличением выборки. Ограничение - необходимость доступа к распределению вероятностей токенов, но общие принципы адаптируемы для любого LLM-интерфейса через многократную генерацию и отбор.

## Ключевые аспекты исследования: 1. **Self-Certainty метрика:** Исследование предлагает новую метрику "Self-Certainty", которая измеряет уверенность LLM в генерируемых ответах, основываясь на дивергенции Кульбака-Лейблера между предсказанным распределением токенов и равномерным распределением. Это позволяет оценивать качество ответов без внешних моделей вознаграждения.

**Borda-Voting с Self-Certainty:** Авторы разработали метод голосования, использующий ранжирование ответов на основе Self-Certainty, что позволяет улучшить выбор из нескольких сгенерированных вариантов.

**Best-of-N selection:** Исследование демонстрирует, что Self-Certainty эффективно масштабируется с увеличением числа сгенерированных ответов (N), что делает метод Borda-Voting с Self-Certainty более эффективным, чем простое мажоритарное голосование или Universal Self-Consistency.

**Применимость к открытым задачам:** Метод Self-Certainty работает с открытыми задачами, где традиционные методы вроде Self-Consistency неприменимы из-за уникальности ответов.

**Интеграция с Chain-of-Thought:** Self-Certainty дополняет методологию цепочки рассуждений, улучшая результаты рассуждений LLM.

## Дополнение:

### Применимость методов исследования в стандартном чате

Исследование "Scalable Best-of-N Selection for Large Language Models via Self-Certainty" фактически **не требует дообучения или специального API** для базового применения его концепций. Ключевой метод Self-Certainty технически использует распределение вероятностей токенов, доступ к которому есть во время инференса, но основная идея может быть адаптирована для стандартных чатов.

### Концепции и подходы для стандартного чата:

**Множественная генерация с последующим выбором:** Пользователи могут запросить модель сгенерировать несколько вариантов ответа на один вопрос. Затем выбрать наиболее последовательный или обоснованный ответ.

**Альтернативы Self-Certainty:**

Запрос к модели оценить свою уверенность в каждом ответе. Использование специальных промптов для выявления ответов, в которых модель наиболее уверена.

**Адаптация Borda-Voting:**

Генерация нескольких ответов и их ранжирование по "качеству рассуждения". Использование частоты повторяющихся ответов как признака их правильности.

**Комбинирование с Chain-of-Thought:**

Запрос к модели рассуждать шаг за шагом и генерировать несколько таких цепочек. Выбор цепочки с наиболее последовательным рассуждением. ### Ожидаемые результаты: - Повышение точности ответов на сложные вопросы - Снижение количества "галлюцинаций" и ошибок в рассуждениях - Улучшение качества генерации кода и решения математических задач - Более надежные ответы на открытые вопросы, где нет единственного правильного ответа

## Анализ практической применимости: 1. **Self-Certainty метрика:** - Прямая применимость: Высокая. Пользователи могут использовать эту метрику для оценки уверенности модели в своих ответах без необходимости внешних инструментов. - Концептуальная ценность: Значительная. Помогает понять, что распределение вероятностей токенов содержит важную информацию о качестве ответа. -

Потенциал для адаптации: Высокий. Концепция может быть адаптирована для различных сценариев использования LLM, включая определение, когда модель "блефует".

**Borda-Voting с Self-Certainty:** Прямая применимость: Средняя. Требуется генерации нескольких ответов и их оценки, что сложнее для обычного пользователя. Концептуальная ценность: Высокая. Демонстрирует, как комбинирование частоты ответов и их уверенности повышает качество выбора. Потенциал для адаптации: Средний. Может быть адаптирован как метод выбора в интерфейсах, предлагающих несколько вариантов ответов.

#### **Best-of-N selection:**

Прямая применимость: Средняя. Обычные пользователи редко генерируют множество ответов, но интерфейсы могут автоматизировать этот процесс. Концептуальная ценность: Высокая. Показывает, что множественная генерация с умным отбором значительно улучшает результаты. Потенциал для адаптации: Высокий. Может быть встроен в интерфейсы как "продвинутый режим" генерации ответов.

#### **Применимость к открытым задачам:**

Прямая применимость: Высокая. Позволяет оценивать качество открытых ответов, где нет "правильного" ответа. Концептуальная ценность: Очень высокая. Расширяет понимание возможностей оценки качества креативных или открытых задач. Потенциал для адаптации: Высокий. Применим для творческих задач, написания кода, дизайна.

#### **Интеграция с Chain-of-Thought:**

Прямая применимость: Средняя. Требуется понимания обоих методов. Концептуальная ценность: Высокая. Демонстрирует синергию между методами рассуждения и оценки уверенности. Потенциал для адаптации: Высокий. Может быть интегрирован в промпты, направляющие рассуждения LLM.

### **Prompt:**

Использование Self-certainty в промптах для GPT ## Основная идея исследования  
Исследование представляет метод **Self-certainty** для выбора лучшего ответа из нескольких вариантов (Best-of-N selection) в больших языковых моделях. Ключевое преимущество — возможность повысить качество ответов без использования внешних моделей вознаграждения.

## Пример промпта с применением знаний из исследования

[=====] # Задача решения математической проблемы с применением Self-certainty

## Инструкция Я хочу, чтобы ты решил следующую математическую задачу. Для

повышения точности: 1. Предложи 5 различных подходов к решению (используя Chain-of-Thought для каждого) 2. Для каждого подхода укажи, насколько ты уверен в правильности решения (от 1 до 10) 3. Выбери решение с наивысшей уверенностью, или если несколько решений имеют одинаковый ответ, выбери то, которое имеет наибольшую поддержку (как в методе голосования Borda)

## Задача Найди значение выражения:  $(3^4 \times 5^2) \div (3^2 \times 5^3)$

## Формат ответа Подход 1: [решение с рассуждением] Уверенность: [оценка] Ответ: [результат]

...

Подход 5: [решение с рассуждением] Уверенность: [оценка] Ответ: [результат]

Итоговый выбор: [выбранный ответ с обоснованием] [=====]

## Как работают знания из исследования в этом промпте

**Множественные решения:** Промпт запрашивает несколько вариантов решения (N=5), что соответствует методологии Best-of-N selection из исследования.

**Chain-of-Thought (CoT):** Каждое решение должно содержать цепочку рассуждений, что улучшает качество ответов согласно исследованию.

**Оценка уверенности:** Запрос уровня уверенности для каждого решения имитирует метрику Self-certainty — модель должна оценить, насколько она уверена в каждом решении.

**Метод голосования:** Выбор ответа с наивысшей уверенностью или с наибольшей поддержкой (если несколько решений дают одинаковый ответ) имитирует метод голосования Borda из исследования.

Такой подход позволяет получить более точный ответ за счет: - Генерации нескольких вариантов решения - Структурированного рассуждения (CoT) - Оценки уверенности модели в каждом решении - Выбора наиболее надежного ответа на основе комбинации уверенности и согласованности

Этот метод особенно эффективен для сложных задач рассуждения, математических задач и задач программирования.