

# Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

## Суть работы

Для текстовых задач используем few-shot обучения, но в промпте вместо коротких ответов подробно расписываем решение. Такой подход дает значительное преимущество на больших языковых моделях. Авторы получили новый state-of-the-art результат на датасете GSM8K, обогнав модель, полученную с помощью файнтьюнинга. Получается 56.5 на PaLM.

## Контекст появления и авторы

На arXiv.org статья появилась в январе 2022 года. В первой версии еще не было sota результата, он появился только к апрелю во второй версии с добавлением PaLM модели на 540 миллиардов параметров. Статья была подана на NeurIPS 2022. Одним из недостатков рецензенты на openreview указывали отсутствие экспериментов с открытыми и бесплатными моделями, поэтому в 5 версии авторы добавили эксперименты на бесплатной модели Codex от OpenAI. С помощью этой модели они побили свой предыдущий рекорд на GSM8K. Всего вышло 6 версий статьи, последняя в январе этого года.

У статьи 9 авторов, все они работают в Google Research. Среди них можно выделить троих, которые продолжили выпускать темы по CoT. Это Jason Wei, Xuezhi Wang и Denny Zhou. Все они давно занимаются NLP и входят в соавторы статьи про модель PaLM, с помощью которой удалось получить sota результат.

## Предшественники и конкуренты

Авторы в основном вдохновлялись статьями по few-shot и zero-shot обучению. Среди них:

1. *The Power of Scale for Parameter-Efficient Prompt Tuning (Lester et. al)*

В этой статье авторы пытаются обучать промпты. Промптом в данном контексте называются эмбединги, которые конкатенируются к запросу. Таким образом получается, что нужно дообучать не всю модель, а только эти эмбединги.

2. *Finetuned Language Models are Zero-Shot Learners (Wei et. al)*

Это предыдущая статья Jason Wei. Авторы дообучают модель на большом числе датасетов, в которых входные данные – инструкция, где написано что модели надо делать. (Перевести на китайский предложение, ответить на вопрос про жанр фильма и т.п.). Оказывается, что у дообученной модели получаются хорошие zero-shot способности на данных, которые она не видела.

Основным нововведением по сравнению с предыдущими работами является подход с изменением не входа, а выхода модели.

## Продолжения

У данной статьи много интересных продолжений:

1. *Self-Consistency Improves Chain of Thought Reasoning in Language Models (Wang et. al)*

Это самое прямое продолжение статьи. С очень похожим составом авторов. На постере CoT статьи к NeurIPS 2022 также фигурировали результаты этой статьи. Возьмем CoT, но теперь будем смотреть на несколько промптов для каждого примера, смотреть на ответы и брать тот, что встречается чаще всего. 74.4 на GSM8k.

2. *Large Language Models Can Self-Improve (Huang et. al)*

Это статья – продолжение статьи про CoT и предыдущей. Итак, для каждого примера мы начали семплировать несколько решений, давайте дообучим модель на них! Получим 82.1 на GSM8k.

3. *On the Advance of Making Language Models Better Reasoners (Li et. al)*

Прокачиваем метод с семплированием нескольких кандидатов. Теперь вместо того, чтобы выбирать ответ, который встречается чаще всего, будем обучать классификатор, который говорит насколько решение похоже на правильное. Получаем 83.2 на GSM8k.

4. *PAL: Program-aided Language Models (Gao et. al)*

Авторы поняли, что чтобы составить правильное решение для задачи, модели необходимо сделать две вещи: первое – составить формулы, по которым должен быть получен результат, второе – подставить значения из задачи в формулы в результат.

Кажется, что вторая часть довольно простая, но на самом деле это не так.

Хорошие способности по арифметическим действиям появляются у моделей только очень большого размера. В статье про CoT 8% процентов ответов были неверными из-за ошибок в вычислениях.

Поэтому в этой статье предлагается генерировать модели не слова, а код на питоне. Получаем 72.0 на GSM8k, что больше, чем у просто CoT.

## Сильные стороны

1. Актуальный способ улучшить качество, так как LM идут в сторону увеличения параметров, а метод как раз хорошо работает для больших моделей.
2. Корректные результаты, сравнение с SOTA моделями.
3. Простой метод, ничего дообучать не надо.

## Слабые стороны

1. Недостаточно исследован prompting и влияние на результат
2. Метод дает профит только на больших моделях, которые доступны далеко не всем.

## Возможные улучшения

1. Попытайтесь увеличить качество на маленьких моделях
2. Эксперименты с prompting'ом. В статье сравниваются промпты, написанные двумя разными людьми. Хотя и написанные разными людьми, промпты бывают весьма похожи. Хочется какую-то семантическую разницу.

3. (Не по статье, а в целом) Кажется, что можно скрестить второе и третье продолжение, то есть дообучить на сгенерированных же ею данных и сделать более хитрый выбор ответа, чем выбор того, кто встретился наибольшее число раз. На такой модели есть шанс получить еще больше на GSM8k.