**Отчет об экспериментах**

В рамках проекта было выполнено несколько экспериментов. Во-первых, проведены заявленные эксперименты с CoT и ансамблированным CoT (к сожалению, получилось только на 3 путях размышлений, а не на 40 как это было в препринте). Получены соответствующие метрики около 13 и 11 процентов соответственно. Оба алгоритма значительно улучшают результаты бейзлайна, который был также проведён, и чья точность без CoT составила 6%. Любопытно, что ансамблирование с той же моделью привело к незначительному ухудшению результата. В результате изучения собранных данных, я пришел к выводу, что на таком количестве параметров модель BLOOM практически неспособна к консистентному выведению ответов, во всяком случае при генерации 3 ответов на одну и ту же задачу. При чем весьма часто проблема в том, что модель систематически ошибается в арифметике, даже если она дает верные пути размышления (более того, зачастую она даёт систематически верные пути размышления и также систематически ошибается в арифметике, что ведёт к неверным ответам). Бейзлайн и CoT результаты вполне сравнимы и практически эквивалентны сравнимым моделям с таким же количеством параметров из предоставленных препринтов в частности с моделью LaMDA со 137 миллиардами параметров. При этом Bloom ощутимо проигрывает GPT со 175 миллиардами в качестве.

Предположительно, падение качества на ансамблированном CoT вызвано тем, что получилось просимулировать только на 3 сэмплах, а нужно существенно больше для того чтобы достичь self-consistency. Очевидно, что это не согласуется с результатами в статье, где было получено улучшение. В целом, в том числе, как и следует из препринта и экспериментов, наши результаты достаточно неустойчивы к случайности и требуют большого количества экспериментов для получения robust результата.

В процессе у меня появилось несколько идей для улучшения результатов.

1). Ансамблирование, но не на уровне декодера, а на уровне ансамбля моделей с разной температурой. Идейно - часть моделей будет более консервативна, часть менее, возможно мажоритарным голосованием мы сможем получить хороший результат. (На идею натолкнулся, когда игрался с одной из задач, получил исправление арифметической ошибки при другой температуре).

2). Классификация/кластеризация задач на входе. Доложим ещё одну модель классификации/кластеризации задач. На обучении мы с помощью классификатора разбиваем задачи и их решения на группы максимально схожих. Затем мы сохраняем показательный пул задач + CoT + решение для каждого класса. После этого когда на вход будет поступать задача будем делать prompt следующим образом, пусть мы классифицируем эту задачу классом 1. Тогда берём соответствующие этому классу задачи с CoT и решением и записываем в prompt. Идея - наиболее подходящая форма цепочки размышлений поможет найти соответствующую цепочку нашей модели. Проблема в том, что классификация требует дополнительных ресурсов на разметку, а кластеризация может дать не лучшие результаты. Тем не менее было бы интересно попробовать.

3). Структурирование запросов. Возможно нам стоит придать больше формальной структуры запросу? Пусть это будет не CoT, а система вида часть предложения -> шаг решения в виде уравнения. Например, "У Васи было 5 яблок, 1 он съел -> 5-1=4". Буквально разбивая на пошаговые уравнения и возможно нумеруя их.

4). Изменение количества показанных решённых задач в промпте. Возможно при few-shot работе, у нас оказывается, что, если мы даём много задач мы "переобучаемся", а если слишком мало "недообучаемся".

Я опробовал на маленьких выборках 1 и 4 подходы, и они дают значительные улучшения на этих 100 примерах из тестировочного датасета вплоть до 14 процентов, а если их тоже ансамблировать, то и до 16. Однако пока что сложно судить о том насколько это устойчивые результаты, стоит тестировать на большем количестве задач, однако это требует слишком больших затрат времени при моих мощностях.

Также интересно, что очень тяжело с решением задач на большое количество действий. Основные улучшения в качестве получаются за счёт улучшения качества на несложных задачах в 2-3 действия. Это даёт нам ещё одну задачу на будущее, в рамках которой предстоит понять, как построить запрос или обучить модель чтобы получилось обобщить задачу на большое количество действий.