**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 для всего тестового датасета, и на трети датасета с 20 путями размышлений). Получены соответствующие метрики около 13 (CoT), 11 (CoT ensembling 3) и 15 (CoT ensembling 20) процентов соответственно. Оба алгоритма значительно улучшают результаты бейзлайна, который был также проведён, и чья точность без CoT составила 6%.

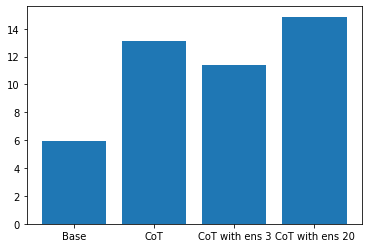


Рис. 1. Сравнение качества полученных моделей в процентах.

Любопытно, что ансамблирование с той же моделью привело к незначительному ухудшению результата при малом количестве путей размышления и к улучшению при большом количестве путей. В результате изучения собранных данных, я пришел к выводу, что модель BLOOM практически неспособна к консистентному выведению ответов при генерации 3 ответов на одну и ту же задачу. На стандартных параметрах генерации слишком велик разброс ответов, что существенно мешает улучшению результатов. При этом мы можем наблюдать сходимость консистентности при увеличении количества сгенерированных путей до 20. 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.

Падение качества на ансамблированном CoT вызвано тем, что получилось просимулировать только на 3 сэмплах, а нужно существенно больше для того чтобы достичь self-consistency. При этом на 20 сэмплах качество растет. 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. Все это в целом согласуется с результатами, полученными исследователями на других моделях, хоть наши цифры и существенно более консервативны и не дают такого же большого увеличения результатов.

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.

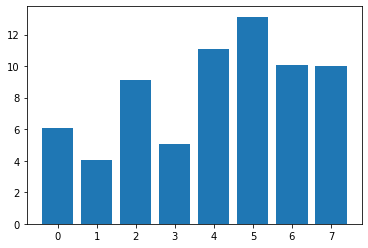


Рис. 2. Колебания качества в процентах на первых 100 задачах в зависимости от количества шотов примеров в промпте CoT.

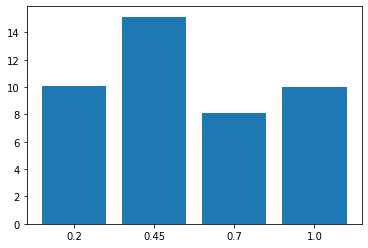


Рис. 3. Колебания качества в зависимости от уровня температуры на первых 100 задачах с CoT.

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. Более того, основным источником улучшения каечтсва моделей с ансамблированным CoT достигается именно на задачах с малым количеством действий как видно это по анализу распределения решений задач. Из-чего следует, что модель добивается консистентности на более простых задачах, потенциально теряя в качестве на более сложных задачах, которые ранее, в силу высокой дисперсии ответов могла решить «случайно» попав в ответ. 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.