Bloom CoT

Отчёт сделал: Ширнин Александр
Моя почта: alexamailwork@gmail.com
GitHub с решением: https://github.com/25icecreamflavors/CoT-ensembles

1. Мои гипотезы

Мы работаем с датасетом GSM8K, в нём содержится примерно 10к простых математических задач, где требуется сделать пару действий, чтобы получить численный ответ. В датасете есть трейн и валидация. Из трейна мы можем взять несколько решений для промптов, а на валидации мы прогоняем эти промпты и считаем ответы. Из него я сделал свой датасет, где сделал несколько вариантов промпта для CoT, а также промпты под свои гипотезы, а дальше вставил туда вопросы из теста. Ссылка на ноутбук:
https://www.kaggle.com/code/manwithaflower/prompts-dataset-preparation

Сразу скажу, что я сделал датасет под свои гипотезы-промпты и загрузил его в последний день перед первым дедлайном на kaggle, так как я там использовал расчёты. Я думал, что уже конец, а также думал, что никому не нужен мой kaggle. Но на этой неделе я внезапно увидел, что внезапно его скачали больше 5 раз. Поэтому, если вы у кого-то в решении увидете прям такую же гипотезу, с большой вероятностью они скатали у меня. По датам можете проверить, что я загрузил первым (gotcha).

После чтения статей сразу становится понятно, что есть куча комбинаций, которые можно пробовать для экспериментов. На качество может влиять и порядок задач в промтах, и их количество, и то как написаны решения (навык аннотаторов). При этом авторы пишут, что после 8 задач качество примерно становится одинаковым. Но от аннотаторов качество зависит сильно. При этом, если мы используем ансемблированный CoT, то качество, конечно, может ещё и зависеть от гиперпараметров. В целом, авторы проверили много параметров и написали, что такое решение достаточно стабильное (robust), там есть некие колебания в качестве, но они не такие существенные, то есть перебрать несколько параметров и найти более удачный - было не так сложно. Я считаю, что в идеале для хорошей статьи следует: запускать модели на нескольких сидах (для стабильности и уменьшения случайности), на оптимальном количестве промптов (8-16), а также использовать рекомендуемые параметры сэмплинга. Насчёт аннотаторов - мне кажется, что тема немного спорная. С одной стороны, подобрать лучшего по тесту - выдаст лучший результат. С другой стороны, в каком-то смысле тут может выходить переобучение под тест. Возможно, стоит смешивать решения аннотаторов или ансамблировать сгенерированные ответы. Перебирать гиперпараметры для повышения качества кажется немного скучным и непрорывной идеей (к авторам статьи была такая претензия насчёт промптинга), поэтому это я бы не стал делать.

Самым интересным вариантом улучшения качества кажется придумать новый вариант промптов, пайплайн ансмаблирования. Когда я придумывал идеи решения, я опирался на текущие проблемы в решениях по своему видению. Во-первых, модели часто не используют нормальный счёт, то есть они генерируют вычисления, но по факту “не пользуются калькулятором”. Во-вторых, когда делают анэсмаблирование, между этими решениями нет связи, они получены независимо, можно сказать. А также есть проблема, что модели показывается единственный вариант верного решения в промпте. Ниже опишу свои гипотезы:

1. Direction CoT:
   Мне хотелось сделать акцент модели на том, чтобы она использовала математические операции. А также хотелось добавить фразу “объяснение решения”, чтобы она понимала, что ей следует не просто написать кратко ответ, а аргументировать. Я думал, что так она будет рассуждать и вероятность получить верный ответ будет больше.

   ```
   Use mathematical operations and write step by step the explanation of the following math problem solution:
   Q: Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for $2 per egg. How much does she make every day?

   Here is the solution explanation:
   A: She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = $18 per day. The answer is $18.

   Use mathematical operations and write step by step the explanation of the following math problem solution:
   Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

   Here is the solution explanation:
   ```

   Но при небольшом тесте были некоторые проблемы. По каким-то причинам после последней фразы она иногда начинала просто повторять вопрос, а не объяснять решение. Также она пропускала букву “A:”. Предположу, что либо в конце следует удалить фразу о решении, либо надо добавить больше задач в промпт.

2. CoT paths:
   Как я писал выше, я хотел модели дать понимание, что задачу можно решить несколькими способами, как минимум, решение можно описать разными словами. И в ответ на это хотелось получить сразу несколько ответов, которые можно было бы ансамблировать. Ниже приведу пример промпта:

```
Q: Janet’s ducks lay 16 eggs per day. She eats three for
breakfast every morning and bakes muffins for her friends every
day with four. She sells the remainder for $2 per egg. How much
does she make every day?
A: She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = $18
per day. The answer is $18.
A: She eats 3 for breakfast, so she has 16 - 3 = 13 left. Then
she bakes muffins, so she has 13 - 4 = 9 eggs left. So she has 9
eggs * $2 = $18. The answer is $18.
A: This means she uses 3 + 4 = 7 eggs every day. So in total she
sells 16 - 7 = 9 eggs. She sells the remainder for $2 per egg, so
she makes 9 * $2 = $18 per day. The answer is $18.


Q: If there are 3 cars in the parking lot and 2 more cars arrive,

how many cars are in the parking lot?
```

Тут тоже возникли некоторые проблемы при генерации. В промпте написано 3 решения, но модель выдавала случайно количество: иногда 1, иногда 2, а после этого начинала выдумывать следующую задачу “Q:” и решать уже её. Но вот эта идея мне казалась классной и перспективной. Как-то хотелось бы её доработать, чтобы модель могла видеть свои несколько рассуждений и выбирать среди них лучшее. Хотя самый простой вариант - это просто делать majority vote от нескольких её ответов в таком варианте.

2. Проверенные технические решения (запуск моделей)

К сожалению, проблемы с запуском экспериментов начались сразу после генерации идей. У меня сейчас нет доступных мощных ресурсов, поэтому мне оставалось использовать только ноутбуки google colab или kaggle. При этом kaggle лучше, потому что там ноутбуки можно запускать на несколько часов, чтобы велись расчёты, а google colab часто сбрасывает сессии.

В самом начале я попробовал использовать предложенную модель от petals. С ней были одни проблемы. Иногда просто не работало подключение, иногда генерация останавливалась в середине процесса. В лучшем случае работала генерация, но один сэмпл прогонялся примерно минуту, при этом я ставил max_new_tokens=5. Авторы предложенных статей писали, что они у всех моделей ставили этот параметр равным 128. Когда я пытался увеличить длину, то генерация просто не работала, она зависала в каком-то моменте и ответ никак нельзя было получить. В общем, меня крайне разочаровало это решение, модель совсем не работает в моём случае.

Задание, конечно, хотелось выполнить, потому что были интересные гипотезы, а также сама тема мне нравится. Следующей попыткой было попробовать bloom с HF, просто взять модель меньше. Была надежда, что модель запустится и что она может показать какой-то неожиданный результат, что в теории можно было бы сказать:"Ух ты, мы проверили модель меньше размером, а она выдаёт неплохой результат". К сожалению, проблемы были и тут: у меня никак не запускались модели на 7 и 3 миллиарда параметров. Видимо, недостаточно памяти, я пробовал обходные пути, ставил параметры device_map, low_cpu_memory. Но, к сожалению, вечно были какие-то ошибки по памяти итд, пришлось оставить эти модели. Заработала модель на 1.7 миллиарда параметров. Сэмплы прогонялись, но результат разочаровал. Модель просто долго повторяла вопросы, вместо генерации ответов. Интересно, что при некоторых промптах модель что-то начинала пытаться считать и генерировать, но это было редко, а также совершенно рандомно и неаргументированно. В ноутбуке можно посмотреть несколько примеров.
Ссылка на код и запуск: https://www.kaggle.com/code/manwithaflower/bloom-1b7-gsm8k

У меня больше не осталось вариантов как-то это запускать, поэтому я решил в конце попробовать hugging face api. Это было бы неплохим решением, потому что, как оказалось, в запросе можно даже передавать параметры (даже сэпмлпинга). И в целом генерация работала достаточно быстро, у меня генерировался сэмпл не 1 минуту, а секунд 5-10. Но, к сожалению, оказалось, что по бесплатному доступу там ограниченное число запусков. Поэтому я пару часов писал код, а потом выяснилось, что прогнать даже 100 сэмплов с помощью разных способов я не могу, что меня опять же довольно-таки расстроило. Но в целом я написал код, с помощью которого прогоняются промпты и из них вынимается числовой ответ (часть заимствовал из приложенных материалов авторов статьи). Одной из проблем было то, что генерируется недостаточно длинный ответ, поэтому я продолжал подавать сгенерированный текст, пока не удастся изъять ответ из генерации. При этом я поставил ограничение на 20 запросов, чтобы при ошибках процесс не зацикливался (а изредка они были). Под ошибками я имею в виду, например, ситуацию, когда модель по каким-то причинам начинает повторять один и тот же текст. Также интересный факт, если перед "Q:" или "A:" делать красные строки, то есть "\n" в питоне и в таком виде подавать текст, то генерация не работает, модель выдаёт обратно исходный текст. Но если подавать с красными строками текст на сайте HuggingFace, то там генерация работает. В целом удалось всё-таки прогнать несколько сэмплов, посмотреть глазами на ответы, поэтому некоторые мысли и гипотезы есть.
Ссылка на код и запуск (не обращайте на ошибку в конце ноутбука, я остановил прогон, когда увидел, что **токен умер**):
https://www.kaggle.com/code/manwithaflower/bloom-hf-api-gsm8k

3. Выводы

Это очень интересная задача, мне понравилось придумывать гипотезы по этой теме. Для хорошей статьи стоит решить вначале проблему с моделью. Возможно, стоит использовать другие модели, если эта никак не запускается. Чтобы идея была новой, стоит попробовать другие варианты промптинга, не уверен, что просто ансамблирование по различным гиперпараметрам, сидам, аннотаторам понравится

ревьюверам. При этом обычное ансмаблирование с сэмплированием авторы уже хорошо протестировали. Кажется, что следующий шаг - это ансамбль внутри одного ответа модели (что-то похожее на мой вариант), а затем, возможно, стоит вот эти несколько решений подавать ещё раз в генеративную модель (возможно, другую), чтобы она среди них выбрала верное. Но всё ещё существует проблема с вычислениями и с неким отсутствием аргументацией.