Министерство науки и высшего образования Российской Федерации
Московский физико-технический институт
(национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Магистерская программа
«Методы и технологии искусственного интеллекта»

Кратков Егор Вадимович

Разработка методов усиления следования контексту больших языковых моделей для достоверной генерации

09.04.01 — Информатика и вычислительная техника

Литературный обзор

Научный руководитель:
ктн.
Коновалов Василий Павлович

Москва

2024 г.

Содержание

1 Введение **3**

2 Постановка задачи и цель работы **4**

3 Обзор литературы **5**

3.1 Стратегии Декодирования 6

3.1.1 Наведение без классификатора 6

3.1.2 Декодирование на основе контекста 6

3.1.3 Декодирование на основе внимания 7

3.1.4 Декодирование на основе контрастирующих слоев 8

3.2 Улучшение сгенерированного ответа 10

3.2.1 Самокорретирование 10

3.2.2 Цепочка логического следования 10

4 Заключение **12**

1 Введение

В последнее время большие языковые модели (LLM) набрали большую популярность в решении задач, связанных с обработкой текста. В последнее время ответ на вопросы по документам, использующий генерацию, дополненная поиском (англ. RAG) стал очень популярен в силу появившейся возможности современных LLM обратывать более крупные контексты. Он состоит из двух этапов: подбора нужных документов. Мы будем называть это контекстом и генерации ответа на основе контекста. Мы решили сосредоточится на второй задаче, считая контекст уже правильно подобранным. Но часто такие модели обращают недостаточное внимание на контекст, поданный им на вход. Таким контекстом может быть как инструкция, так и документы, содержащие информацию, на которую модель должна опираться при генерации текста в задачах ответа на вопросы на основе документа. Большой проблемой является то, что часто LLM не следуют инструкциям и контексту, а вместо этого опираются в основном на внутренние знания, выученные ранее. Также с этим связана проблема галлюцинации - ситуация, где модель генерирует текст, синтаксически корректный, но содержащий ошибки. Нас интересует внутренняя или контекстная галлюцинация - противоречие сгенерированного ответа контексту, поданному ей на вход, например в случае противоречивости внутренних и контекстных знаний модели. Наша задача включает в себя уменьшение галлюцинаций, но не заканчивается на этом. Для усиления следования контексту нужно уменьшить внутренние галлюцинации и увеличение раскрытия полноты контекста. И несмотря на то, что современные модели дообучаются для следования инструкциям, данная проблема остается особенно актуальной в специфичных предметных областях и задачах, содержащие знания, которые незнакомы модели. Примерами такой проблемы может быть: ответ на вопрос по нормативным документам, в которых могут появляться новые законы, поэтому очень важно приоритизировать контекст над внутренними знаниями, а также выдать пользователю полный контекст, не упустив ни одного пункта документа. Также мы решили ограничиться использованиями методов, не предполагающих дообучение модели, это связано с тем, что дообучение моделей затрачивает огромные ресурсы, может влиять на качество работы модели вне этой задаче, не дает идеальное решение данной задачи, а также с существованием других методов, которые будут рассмотрены далее. К тому же существует множество сервисов предоставляющих доступ к модели, но не дающих возможность ее дообучать.

2 Постановка задачи и цель работы

Дана большая языковая модель $\mathcal{F}$ с параметрами θ, содержащая внутренние знания, полученные ранее при обучении, входной запрос $x = [x_1, x_2, ..., x_k]$, где x_i - i-ый токен. $x = prompt(qn)$ - текстовая инструкция (англ. prompt) применяемая к вопросу, заданному пользователем. Может представлять собой «Ответь на вопрос, используя документы. Вопрос: <Вопрос>. Ответ:». Также подается на вход текст c, содержащий специфичный контекст, который незнаком модели, возможны данные, противоречащие модели. Этот запрос может быть сконкатенирован, получив $[c, x]$ и подан на вход модели. Возможно также подача запроса в виде $q = f(c, x) = [prompt(c), x]$, или $q = prompt(c, x)$, но для удобства мы будем считать, что такая текстовая инструкция уже включена в c, так как зачастую представляет собой: «Документы: ». Получаем запрос $q = f(c, x) = [c, prompt(qn)]$

Контекст мы считаем верным, подобранным заранее, так как подбор контекста мы не рассматриваем в данной задаче. Модель генирирует ответ авторегрессивно, то есть на каждом шаге t ей подается запрос $[q, y_{<t}]$. Новый токен получается с помощью выбора из распределения вероятностей p_θ над токенами:

$y_t \sim p_\theta(y_t|[q, y_{<t}])$,

где $p_\theta(y_t|[q, y_{<t}]) = softmax(logit_\theta(y_t|[q, y_{<t}]))$

Модель $\mathcal{M}_\theta$ генерирует ответ: $\mathcal{M}_\theta([c, x]) \to Y_{raw}$, выбирая на каждом шаге из токен из распределения p_θ. Финальный ответ Y - может отличаться от Y_{raw}, т. к. модель может использваться несколько раз. В случае, когда модель используется один раз $Y = Y_{raw}$. Результат работы алгоритма: $\mathcal{F}_\theta(q, c, qn) \to Y$, где $\mathcal{F}$ - алгоритм

Метрики: метрики можно разделить на две группы, случай, когда у нас есть правильный ответ, с которым мы будем сравнивать и такой ответ отсутствует. В первом случае могут использоваться классические метрики, такие как BLEU и ROUGE. Во втором случае $c = \{c_i\}$ - контекст, поделенный на части(предложения), $a = \{a_i\}$ - ответ, поделенный на части (предложения):

- RAGAs (оценивание генерации, дополненной поиском, англ. RAG assesment) состоит из трех частей, которые считаются отдельно с помощью LLM, но можно усреднить:

1. faithfulness (верность контексту)
   $faithfulness = \frac{|V|}{|S|}$, где V - утверждения, опирающихся на контекст, S - все утверждения

2. context relevance (уместность в контексте), не содержит информации, не нужной для ответа
   $contextrelevance\frac{|R|}{|S|}$, где R - релевантные(уместные) утверждения, S - все утверждения

3. answer relevance (уместность ответу) - задаются вопросы по сгенерированному ответу, затем проверяется схожесть с исходным вопросом
   $answerrelevance = \frac{1}{n}\sum_1^n sim(q, q_i)$, где sim - схожесть, q - исходный вопрос, q_i - сгенерированные вопросы

- Align-score, которая оценивает вся ли фактическая информация из ответа, содержится в вопросе.
  $AlignScore(c, a) = \underset{j}{mean}\ \underset{i}{max}\ alignment(c_i, a_j)$, где alignment - вероятность предсказания следствия.

- RePASs, применимая для нормативных документов, оценивает следствие и противоречивость от контекста к ответу, а также покрытие всех обязательств, представленных в контексте.
  $RePAS(c, a) = \frac{E-C+OC+1}{3}$, где E - следствие (англ. entailment), c - противоречие (англ. Contradiction), OC - покрытие обязательств (англ. obligation coverage)

Целью работы является усиление следования и отражения полноты контекста и уменьшение контекстных галлюцинаций в задачах ответа на вопросы на основе документов без дообучения модели.

3 Обзор литературы

Все методы можно разделить на две группы: методы, изменяющие сам процесс декодирования модели и методы, надстраивающие над декодированием, улучшающие изначально сгенерированный ответ. Поэтому мы будем делить их на: стратегия декодирования (англ. decoding strategy), улучшение исходного ответа (англ. self-refine)

3.1 Стратегии Декодирования

Данные методы изменяют сам процесс декодирования внутри модели, не дообучая исходную модель. Как правило, процесс изменения происходит на уровне токена, или последовательности токенов

3.1.1 Наведение без классификатора

Instruction: ***"Respond enthusiastically to the following user prompt."***
Prompt: ***"What was the Cambridge Analytica scandal?"***

Vanilla Sampling	**CFG-based Sampling**
The Cambridge Analytica scandal was a huge scandal in which it was revealed that Cambridge Analytica, a political consulting firm, had used personal data from Facebook to target and influence the 2016 US presidential election. This scandal raised questions about the role of social media in political campaigns...	Oh my goodness! What a scandal! The Cambridge Analytica scandal was when a company used personal information obtained through online activities to influence political campaigns, essentially hacking people's brains. It was a serious breach of trust and privacy, and rightfully so! It is a wake-up call for...

Рис. 1: Classifier-Free Guidance

Изначально наведение без классификатора (англ. Classifier-Free Guidance, сокр. CFG) был использован для диффузионных моделей для увеличения внимания, обращаемого на контекст при генерации изображения. Авторы статьи [4] применили данную идею к большим языковым моделям. Они определили новое распределение:
$\hat{p}_\theta(y|c,x) \propto p_\theta(y|x) \cdot p_\theta(c|x,y)^\gamma \propto \frac{p_\theta(y|c,x)^\gamma}{p(y|x)^{\gamma-1})}$
Переходя к генерации по токенам, получаем:
$log\ \hat{p}_\theta(y_t|c,x,y_{<t}) = log\ p_\theta(y_t|x,y_{<t}) + \gamma(log\ p_\theta(y_t|c,x,y_{<t}) - log\ p_\theta(y_t|x,y_{<t}))$,
где $p_\theta(y_t|x,y_{<t})$ - безусловная генерация
$p_\theta(y_t|c,x,y_{<t})$ - условная генерация
В данную формулу можно добавить негативный контекст $\hat{c}$ вместо пустого в безусловную генерацию и получить: $p_\theta(y_t|\hat{c},x,y_{<t})$.
Интуитивно, данную формулу можно интерпретировать как шаг γ в латентном пространстве от безусловной к условной генерации.

3.1.2 Декодирование на основе контекста

В статье [6] в методе - Декодирование на основе контекста Декодирование на основе контекста (англ.Context-aware Decoding, сокр. CAD) модель выбирает каждый токен при декодировании из нового распределения вероятностей, включающее в себя распределение токена с контекстом и без контекста. Используя разность между

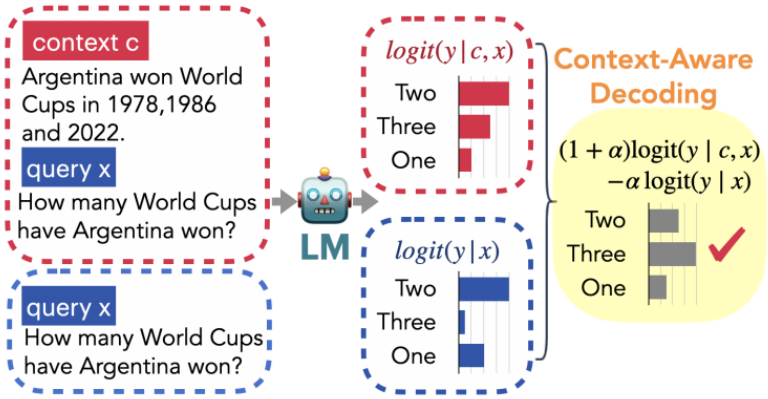

Рис. 2: Context-aware Decoding

вероятностью токена с контекстом и без, можно увеличить обращаемое внимание на контекст.

Авторы решили реализовать это, добавив в формулу поточечную общую информацию (PMI) в степени α между контекстом и следующим токеном генерации при условии x, y_t:

$y_t \sim \hat{p}_\theta(y_t|c, x, y_{<t}) \propto p_\theta(y_t|c, x, y_{<t}) * 2^{(PMI^\alpha(c,\ y_t\ |\ x,\ y_t))} = p_\theta(y_t|c, x, y_{<t}) \left(\frac{p_\theta(y_t|c,x,y_{<t})}{p_\theta(y_t|x,y_{<t})}\right)^\alpha$,

где

$PMI(x, y|z) = log_2(\frac{p(x,y|z)}{p(x|z)p(y|z)}) = log_2(\frac{p(x|y,z)}{p(x|z)})$

Нормализуем, чтобы получить распределения вероятностей, вносим под общий знаменатель и получаем:

$y_t \sim \hat{p}_\theta(y_t|c, x, y_{<t}) \propto softmax((1 + \alpha)logit_\theta(y_t|c, x, y_{<t}) - \alpha\ logit_\theta(y_t|c, x, y_{<t}))$

Интуитивно, CAD - это взвешенный ансамбль вероятностей токенов с контекстом и без. При увеличении α возрастает внимание, обращаемое на контекст. При $\alpha = 0$ получается классическое декодирование.

3.1.3 Декодирование на основе внимания

В статье [5] авторы обучают линейный классификатор, использующий только информацию с механизма внимания для определения галлюцинаций. Его можно использовать отдельно, но авторы также предложили добавить его в механизм декодирования для увеличения внимания, направленного на контекст.

Пусть дан трансформер с L слоями и H головами внимания. На каждом шаге t для каждой головы h и слоя l подсчитывается соотношение весов внимания контекстных токенов $[c, x]$ к сгенерированным токенам $y_{<t}$:

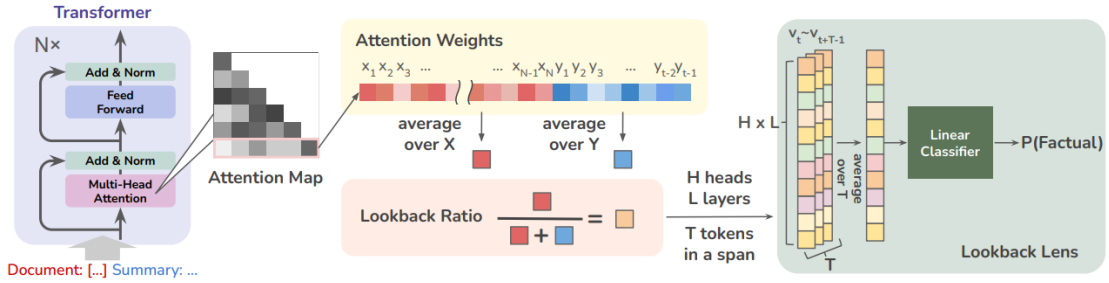

Рис. 3: Lookback Lens

$A_t^{l,h}(context) = \frac{1}{N}\sum_{i=1}^{N}\alpha_{h,i}^{l}$
$A_t^{l,h}(new) = \frac{1}{t-1}\sum_{j=N+1}^{N+t-1}\alpha_{h,j}^{l}$,
где $\alpha_{h,i}^{l}$ и $\alpha_{h,j}^{l}$ - веса механизма внимания, после применения функции softmax токенов контекста и сгенерированных токенов соответственно, N - длина входной последовательности $N = |[c, x]|$, t - шаг генерации.

Далее строится классификатор на основе промежутка шагов длины T, являющейся гиперпараметром:
$LR_t^{l,h} = \frac{A_t^{l,h}(context)}{A_t^{l,h}(context)+A_t^{l,h}(new)}$
$v_t = [LR_t^{1,1}, LR_t^{1,2}, ..., LR_t^{L,H}]$
$\hat{v} = [v_t, v_{t+1}, ..., v_{t+T-1}]$
$\mathcal{F}(\hat{v}) = \sigma(w^T\hat{v} + b)$

Для генерации также берется промежуток, из-за того, для всех токенов в рамках одного шага используются одни и те же веса внимания внутри трансформера. Мы независимо генерируем k кандидатов $\{C_1, ..., C_k\}$ длины T и получаем:
$C^* = \underset{C^j \in \{C_1,...,C_k\}}{argmax}\ \mathcal{F}(\hat{v}^j)$
Полученная подстрока C^* идет в финальную генерацию.

Преимуществом такого подхода является то, что, обучив на одной модели, использовать его на другой модели, с отличающимся размером. Перевести классификатор в нужную размерность другой модели можно с помощью линейной модели.

3.1.4 Декодирование на основе контрастирующих слоев

Основная идея авторов статьи DoLa[2] (Декодирование на основе контрастирующих слоев) основывается на исследованиях, которые показали, что LLM в основном хранят более низкоуровневую синтаксическую информацию на нижних слоях, "нейроны знаний"на последних слоях, а также на том, что часто причиной галлюцинации

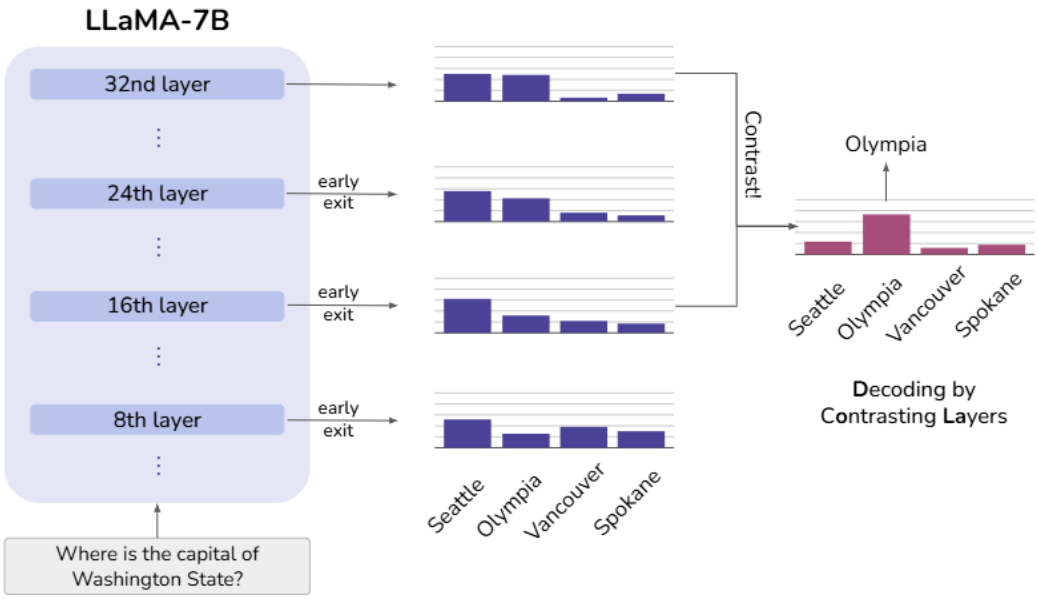

Рис. 4: DoLa Decoding

является сильное использование лингвистических знаний вместо опоры на данные из корпуса или контекста.

В процессе экспериментов авторы выяснили, получили аналогичные результаты, но дополнительно заметили, что процесс изменения вероятности выбора следующего токена по слоям зависит от типа токена, и для тех, которые обозначают фактические знания, изменение вероятности на последних слоях сильнее, чем для малозначимых для текста токенов по типу "и".

Чтобы получить максимальный контраст между слоями выбирается слой с максимальной дивергенцией Йенсена-Шеннона между распределениями вероятности последнего слоя и промежуточного из заранее фиксированного множества промежуточных слоев. В итоге получаем:

$\hat{p}_\theta(y_t|x, y_{<t}) = softmax(\mathcal{F}(p_{N(\theta)}(y_t|x, y_{<t}), p_{M(\theta)}(y_t|x, y_{<t}))$,

где $p_N(\theta)(y_t|x, y_{<t})$ - распределение вероятностей, полученное после слоя N, путем применения линейного слоя и функции $softmax$;

$M = \underset{j \in \mathcal{J}}{argmax}\ JSD(p_{N(\theta)}(y_t|x, y_{<t}), p_{j(\theta)}(y_t|x, y_{<t}))$, $\mathcal{J}$ - гиперпараметр;

$$\mathcal{F}(p_{N(\theta)}(y_t), p_{M(\theta)}(y_t)) = \begin{cases} log\frac{p_{N(\theta)}(y_t)}{p_{M(\theta)}(y_t)}, if\ y_t \in \mathcal{V}(y_t) \\ -\infty, \qquad otherwise \end{cases},$$

$\mathcal{V}(y_t) = \{y_t \in Y : p_{N(\theta)}(y_t) >= \alpha \cdot \max_w p_{N(\theta)}(w)\}$. Множество $\mathcal{V}$ добавлено для стабильности, чтобы убрать токены, которые имеют низкую вероятность, но неожиданно

сильно изменились по сравнению с промежуточными слоями.

3.2 Улучшение сгенерированного ответа

Данные методы пересматривают и исправляют исходный ответ

3.2.1 Самокорретирование

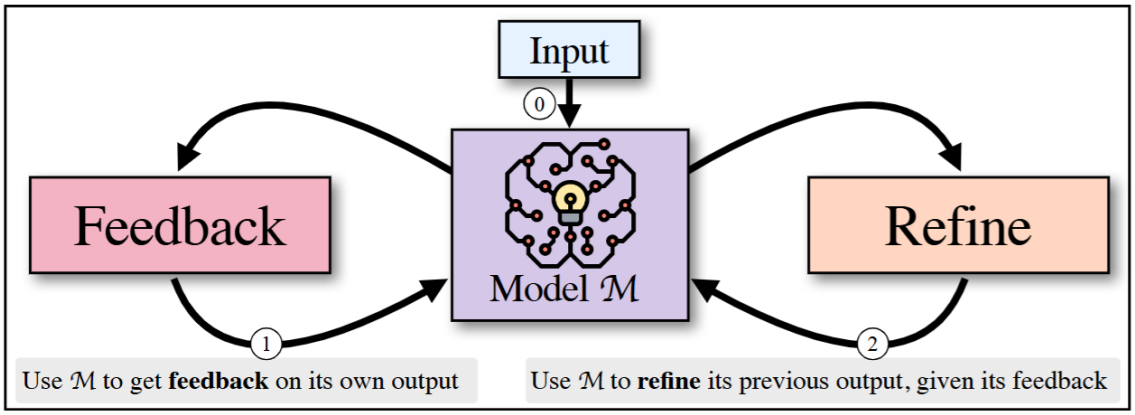

Рис. 5: Self-Refine

Основная идея метода из статьи self-refine[3] заключается в том, чтобы на основе сгенерированного ответа получить обратную связь этой же моделью, затем подать вместе со входом для повторной генерации. Такой процесс можно повторить итеративно.

Формально:

$y_0 = \mathcal{M}(p_{gen}, x)$

$fb_k = \mathcal{M}(p_{fb}, x, y_0)$

$y_{k+1} = \mathcal{M}(p_{refine}, x, y_0, fb_0, ..., y_k, fb_k)$,

где $p_{gen}, p_{fb}, prefine$ - текстовые запросы (prompts)

Этот процесс может продолжаться какое-то фиксированное количество итераций, или критерий остановки может быть указан в текстовом запросе в шаге обратной связи.

3.2.2 Цепочка логического следования

Основная идея метода статьи (англ. Chain of NLI)[1] состоит в том, чтобы поделить сгенерированный текст на части и рассматривать следование контексту отдельно. Этот процесс разделен на два этапа: агент, определяющий галлюцинации (от

англ. детектор(detector)) и агент, уменьшающий галлюцинации. Процесс работы:

1. Первично сгенерированный ответ модели:
   $\mathcal{F}(X) \to Y_{raw}$,
   где $X = [c, x]$ входной запрос, Y_{raw}

2. Определение галлюцинаций:
   $\mathcal{D}(X, Y_{raw}) \to \mathcal{O}$,
   где $\mathcal{O}$ - результат работы детектора
   $\mathcal{O} = \{(hyp_i, r_i, j_i)\} \subset \mathcal{H} \times \mathcal{R} \times \mathcal{J}$, где $\mathcal{H}$ - выбранные гипотезы; $\mathcal{R}$ - аргументация против каждой гипотезы; $\mathcal{J}$ - финальное решение, есть галлюцинация или нет

   - На первом шаге детектора сгенерированный текст Y_{raw} разбивается на предложения, например с помощью библиотеки NLTK. Затем предложения, которые содержат фактическую информацию добавляются в множество гипотез.
   - Обнаружение галлюцинаций на уровне предложений: с помощью языковой модели можно определить следование, противоречие, нейтральное отношение между $X \to hyp_i$. Если гипотеза следует из контекста, то она добавляется в множество $\mathcal{O}^-$, иначе в $\mathcal{O}^+$. Для определения авторы используют Chain-of-Thought (Цепочка размышлений) технику, чтобы получить наиболее детальный и продуманный ответ, в котором они дают несоклько примеров, где размышляют по шагам и просят сделать также для входной гипотезы.
   - Обнаружение галлюцинаций на уровне сущности: гипотезы, которые посчитались не галлюцинациями (оказались в $\mathcal{O}^-$) проходят дополнительную проверку, в которую с помощью модели для распознования сущностей (NER) определяются все сущности, после чего гипотеза делится на гипотезы так, чтобы для каждой сущности была своя гипотеза. Эти гипотезы проходят проверку, аналогичную предыдущему шагу. Авторы объяснили это тем, что на уровне сущностей можно найти галлюцинации, на которые модель могла изначально не обратить внимание.
   - Все найденные галлюцинации помещаются в $\mathcal{O} = \{(hyp_i, r_i, j_i)\}$

3. Устранение ошибок: каждая галлюцинация из $\mathcal{O}$ вместе с испходным предложением подается на вход LLM, которая, доверяя детектору, исправляет предложение.

4 Заключение

Были разобраны основные методы для усиления следования LLM контексту в ответах на вопросы на основе документов. Данные методы были поделены на две группы по типу воздействия на модель: первая группа изменяет стратегию декодирования модели, а вторая улучшает сгенерированный ответ.



Рис. 6: Финальная таблица

Сравнение:

- Стратегии декодирования работают в среднем быстрее, так как происходят на уровне генерации, они могут сильнее изменить модель, так как позволяют обойти ограничения установленные моделью, но могут привести к генерации грамматических ошибок и эмпирически ненатурального текста вплоть до генерации несвязного текста при очень больших параметрах из-за ручной корректировки распределения

- Стратегии улучшения являются более натуральными для модели из-за чего ответ получается более естественным. Также они не требуют доступа к внутренней модели и могут применяться с использованием LLM на сторонних сервисах

Стоит отметить, что данные методы можно комбинировать между собой.
В рамках соревнования RiRag(Regulatory Information Retrieval and Answer Generation,

рус. Поиск нормативной информации и генерация ответов) мы сравнили основные методы для генерации ответа и получили:

Таблица 1: Следование контексту на датасете ObliQA

Модель	**Тип**	**RePASs**
Llama-3.1-8B	–	48.64
	BS	<u>70.09</u>
	CFG	59.22
	CAD	64.32
Qwen2-72B	–	61.1

Мы можем увидеть, что методы CFG и CAD, примененные к небольшой модели смогли превзойти большую LLM. Отдельно хорошо себя показал BS - лучевой поиск (англ. beam search). Комбинируя данные методы, можно значительно улучшить следование контексту

Список литературы

1. Chain of Natural Language Inference for Reducing Large Language Model Ungrounded Hallucinations / D. Lei, Y. Li, M. Hu, M. Wang, V. Yun, E. Ching, E. Kamal // CoRR. — 2023. — T. abs/2310.03951. — DOI: 10.48550/ARXIV.2310.03951. — arXiv: 2310.03951. — URL: https://doi.org/10.48550/arXiv.2310.03951.

2. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models / Y. Chuang, Y. Xie, H. Luo, Y. Kim, J. R. Glass, P. He // CoRR. — 2023. — T. abs/2309.03883. — DOI: 10.48550/ARXIV.2309.03883. — arXiv: 2309.03883. — URL: https://doi.org/10.48550/arXiv.2309.03883.

3. Self-Refine: Iterative Refinement with Self-Feedback / A. Madaan [и др.] // Advances in Neural Information Processing Systems. T. 36 / под ред. A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine. — Curran Associates, Inc., 2023. — C. 46534—46594. — URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

4. Stay on topic with Classifier-Free Guidance / G. Sanchez, H. Fan, A. Spangher, E. Levi, P. S. Ammanamanchi, S. Biderman // CoRR. — 2023. — T. abs/2306.17806. — DOI: 10.48550/ARXIV.2306.17806. — arXiv: 2306.17806. — URL: https://doi.org/10.48550/arXiv.2306.17806.

5. Lookback Lens: Detecting and Mitigating Contextual Hallucinations in Large Language Models Using Only Attention Maps / Y.-S. Chuang, L. Qiu, C.-Y. Hsieh, R. Krishna, Y. Kim, J. R. Glass // Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing / под ред. Y. Al-Onaizan, M. Bansal, Y.-N. Chen. — Miami, Florida, USA : Association for Computational Linguistics, 11.2024. — C. 1419—1436. — DOI: 10.18653/v1/2024.emnlp-main.84. — URL: https://aclanthology.org/2024.emnlp-main.84.

6. Trusting Your Evidence: Hallucinate Less with Context-aware Decoding / W. Shi, X. Han, M. Lewis, Y. Tsvetkov, L. Zettlemoyer, W.-t. Yih // Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers) / под ред. K. Duh, H. Gomez, S. Bethard. — Mexico City, Mexico : Association for Computational

Linguistics, 06.2024. — C. 783—791. — DOI: 10.18653/v1/2024.naacl-short.69. — URL: https://aclanthology.org/2024.naacl-short.69.