RAG における LLM の学習と評価：FAQ タスクへの応用

長澤春希 [1] 戸田隆道 [1]
[1] 株式会社 AI Shift
{nagasawa_haruki,toda_takamichi}@cyberagent.co.jp

概要

大規模言語モデル (Large Language Model: LLM) はその汎用性の高さから，実応用を含めた様々な利活用が進んでいる．昨今では文書などを追加入力として与えることで，外部知識を参照させながら LLM を運用する Retrieval-Augmented Generation(RAG) などの手法の有用性が改めて認識されている．一方で，Low-Rank Adaptation (LoRA) などの軽量な fine-tuning 手法なども確立されつつある．

そこで本稿では，参照知識が限定的な FAQ タスクを例に取り，RAG と比較した際の質問応答タスクにおける LLM fine-tuning の有用性を検討する．並行して，実運用を踏まえた評価についても議論する．

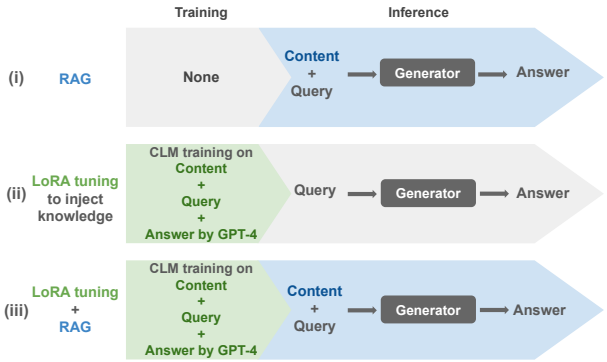

図 1 実験概要図．(i) 追加学習はせず，推論時に質問と関連文書を与えて回答を生成．(ii) 質問と関連文書に加え，それに対する GPT-4 の模範回答を使用して Causal Language Modeling（CLM）にて学習を実施．推論時は質問文のみを入力し回答を生成．(iii) は (ii) で学習したモデルに対し，推論時にも関連文書を与えて回答を生成．

1 はじめに

実応用を含めた様々なタスクで使用されるニューラル言語モデルは，そのパラメータをより大規模化させることで汎化性能が向上することが明らかになりつつある [1]．その結果数百億にものぼるパラメータを有した大規模言語モデル（LLM）の開発が盛んになっている．LLM は学習の過程でテキストデータからさまざまな知識を獲得しているが，これらは有限のパラメータに閉じたものとなっている [2] [3]．従って，学習データに含まれていない知識には対応できないなどの問題点が存在する．そこで外部情報として関連した文書などの情報を入力として与えることで，その情報に基づいた出力を促す Retrieval-Augmented Generation (RAG) [4] [5] などの手法が提案されている．これにより，学習外の情報を利用した推論や，正確な情報の提供による hallucination の抑制などが図られている [6]．

RAG は入力されたクエリに関連する文書を検索する Retriever とそれに基づく回答生成を担う Generator によって構成される．検索対象の文書は任意の系列長 (chunk size) で区切ることにより管理され，この区切り方が後段の文書検索や回答生成の品質に影響すると考えられる．例えばある 1 つの事柄について説明する文書を扱う場合，その事柄について述べられていることを特定するために必要な固有名詞などの情報が全く現れない chunk などが生成される可能性がある．これは必要な情報の検索を難しくする要因の 1 つとなると推察される．このような RAG の設計上の複雑さに由来する難点は各所に存在しており，これらの問題を軽減する 1 つの方法として，chunk に対するアテンションを改善する RETRO [7] などの手法が提案されている．

関連して，LLM の効率的な学習という観点から LoRA [8] や Prompt-tuning [9] などの手法も提案されつつある．具体的には，小規模なパラメータ群を追加して学習させるなどの工夫により，学習コストを抑えながら LLM の学習を実現している．

このような技術を背景とし，関連文書を記憶させるようにモデルを学習させた場合，ある程度まとまったパラメータ内の情報へのアクセスが期待できるため，RAG における先述のような問題を緩和できる可能性も伺える．またビジネス運用などの視点から考えると，学習によって出力分布を意図的に偏らせることによって，競合情報やブランドイメージ

を毀損するようなキーワードの出力や言い回しを回避しやすくなるといったメリットも考えられる．非常に広範なデータで行われる事前学習過程を鑑みると，上述のような fine-tuning の性質は，特定の場面において一定の有用性があると期待される．

そこで本稿では，FAQ タスクにおける LLM の運用を考えた際のモデル学習の有用性を改めて考える．具体的には追加の学習は実施せずに関連文書を質問とともに入力として与える場合（図 1(i)）と，本タスクの文書データを利用して fine-tuning を施し，テスト時には関連文書は与えない場合（図 1(ii)），および追加学習をしたモデルに対してさらに推論時に関連文書を与える場合（図 1(iii)）の 3 つを比較することによって検証を行う．

また本稿のような FAQ タスクでの LLM の評価は，確立された評価方法が存在しないため，本稿では評価手法の選定やその解釈についても議論する．

2 関連研究

RE-PLUG RE-PLUG [10] は Generator のパラメータは固定し，Retriever の学習を行うことにより性能の向上を図っている．具体的には，Retriever の文書検索に係る確率分布と Generator の出力を利用することで，質問文が与えられた時の検索精度の改善を行なっている．

RETRO RETRO [7] は Retriever のパラメータは固定して，Generator の chunk に対する attention の張り方を学習することにより，質問応答タスクなどの性能向上を狙ったものになっている．

RA-DIT RA-DIT [11] は Retriever と Generator の両方の学習を行う．Retriever の学習では KL-Divergence を用いて Generator に適した文書を検索できるようにパラメータを更新する．Generator の学習では，検索文書が与えられた状況下での正解の尤度を最大化するようにパラメータを更新する．

3 性能比較実験：FAQ タスク

3.1 データセット

本稿では株式会社 AI Shift によって作成された Ameba FAQ データセット[1)]を用いて実験を行う．これは株式会社サイバーエージェントが運営する Ameba ブログのヘルプページに掲載されている FAQ データを収集し，これをもとに LlamaIndex にて提供されている Question Generation[2)]などによって拡張されたデータセットとなっている．データセットの構成としては，質問内容を表す Title(Query) とそれに対応するように人手で作成された回答文書の Content(Answer) から成る．この Content から新たな質問を生成するなどのアプローチによって，従来の Ameba ブログには掲載されていなかったものを含めてデータセットを拡張している．また各インスタンスには難易度として easy と hard が設定されており，例えばあるインスタンスに対応する関連文書が学習データに登場する場合は easy，テスト時に初めてその文書が登場する場合は hard として分類されている．本実験ではこの回答文書を質問に対する関連文書として定義し，学習および推論時に利用することとする．また本データセットは学習，評価，テストデータが分離された状態で提供されているため，これをそのまま利用する．ただし今回の検証範囲から，難易度が hard のインスタンスについては，追加学習を施したモデルが回答することは難しいと考えられるため，テストデータでの評価は難易度が easy のもの（687 件）に限定する．

3.2 モデル

本稿では扱うデータセットが日本語であるため，モデルとしても日本語を学習したものを調査対象として選定する．具体的な候補とそのモデルについての情報を Appendix 表 4 にまとめる．またモデルは全て 7B 級の同程度のパラメータサイズのものを利用する．これらのモデルについては学習および評価時に用いるプロンプトとして，HuggingFace の各モデルカードページに記載された推奨プロンプトをそれぞれ使用する．またこれらのモデルに加え，GPT-3.5-turbo および GPT-4 についても比較対象として検証する．なお，この 2 つのモデルについては学習手法等を他のモデルと揃えられないため，学習の対象外とする．

3.3 学習設定

クエリの関連文書に対する学習手法については Low-Rank Adaptation (LoRA) [8] を採用する．学習はモデルの重みを 8-bit 量子化した状態で行い，Ameba FAQ の評価データの学習損失が一定ステップ以上改

1） https://huggingface.co/datasets/ai-shift/ameba_faq_search

2） https://gpt-index.readthedocs.io/en/latest/examples/evaluation/\QuestionGeneration.html#

表 1　Ragas の Answer relevancy（上段）および BERTScore の F1 値（下段）.

Scores	GPT-4	GPT-3.5 turbo	CyberAgent LM	CyberAgent LM2	CyberAgent LM2-chat	ELYZA	ELYZA instruct	StableLM	StableLM instruct
Baseline	0.811 0.673	0.840 0.679	0.718 0.614	0.706 0.594	0.840 0.664	0.715 0.671	0.838 0.669	0.719 0.607	0.805 0.637
RAG	0.829 0.747	0.824 0.731	0.770 0.710	0.767 0.756	0.814 0.721	0.730 0.757	0.804 0.753	0.744 0.809	0.814 0.719
LoRA	- -	- -	0.762 0.646	0.763 0.661	0.814 0.682	0.764 0.670	0.799 0.680	0.744 0.658	0.764 0.659
RAG + LoRA	- -	- -	0.764 0.713	0.769 0.751	0.806 0.741	0.770 0.742	0.795 0.747	0.766 0.745	0.803 0.732

善しなくなるまで継続する．形式としては，系列として質問文と関連文書情報が与えられた際の GPT-4 の出力を模範解答とした時の系列上での causal language modeling (CLM) による学習を考える[3]．

3.4　推論設定

評価時においては，全てのモデルにおいて 8-bit 量子化を行なった状態で推論を行う．また推論時のハイパーパラメータも共通のものを使用し，ビームサーチによるテキスト生成を実施する[4]．この時，モデルによっては 8-bit 量子化の影響で十分なビーム候補を出力できず推論に失敗するインスタンスがいくつか確認された．テストデータ全体に対する割合を鑑み，今回は推論に失敗したインスタンスについては除外して評価を行うこととする．

4　実験結果

4.1　評価指標について

今回は評価指標として Ragas[5] の Answer relevancy と BERTScore [12] を使用する．前者については LLM 自体を評価者として利用する LLM-as-a-judge [13] の手法となっている．また Answer relevancy について，モデルの出力内容によって評価時に弾かれるインスタンスが数件存在したが，先述の方針に則り，このようなインスタンスは除外した上でスコアを算出する．

しかしながら実際に LLM を駆動させながらの評価となるため，スコアの容易な変動や選好バイアスが一定程度存在し，安定した結果の考察が難しいという特徴がある．本稿ではこの事由を踏まえながら，安定したスコア算出が可能である BERTScore も並行して算出し，性能を考察することとした．比較対象は運用種別（図 1 の各種設定）およびモデル種別とした．運用種別に関しては，各設定がどれほど有効かを見るために，追加学習を行っていないモデルに質問のみを入力して回答を生成させる設定についても Baseline として評価を行った．上記スコアによるモデル性能比較の議論が難しい場合については，GPT-4 を評価者とした直接比較によるランキング形式での評価も併せて実施することとした．こちらの評価は，質問と関連文書を鑑みた際に，どの回答が定性的に最も良いかを GPT-4 に自動評価させたものとなっている．ランキングの定量評価としては，情報検索システムの分野で用いられる Mean Reciprocal Rank(MRR) を採用した．これはそれぞれの候補の順位の逆数をスコアとしその平均を算出するものとなっており，取りうる値は 1/n （n は候補数）から 1.0 となる．

4.2　結果と考察

運用種別についての分析　表 1 に各種スコアを掲載している．上段が Answer relevancy の値，下段が BERTScore の F1 値となっている．まず，運用種別における分析（各列での比較）を考えると，ほぼ全てのモデルについて RAG もしくは RAG+LoRA での運用時にいずれかのスコアが最も高くなる傾向が観察された．しかしながら，全体として一貫した傾向が認められにくいことから，この 2 者のどちらがより優れているかをこの結果のみにて結論づける事は難しいため，直接比較における MRR スコアの考察にて議論する（表 2）．また Answer relevancy については，CyberAgent LM2-chat や ELYZA-instruct などにおいて，Baseline 設定でのスコアが最も高くなるという直感に反した結果が観察された．しかしながら Baseline にて実際に生成された回答は Ameba ブログに対する回答というよりも寧ろより一般的なものと

3）　LoRA rank 等の学習時における各種ハイパーパラメータの設定は Appendix 表 5 に掲載.
4）　なお具体的な設定値については Appendix 表 6 に掲載
5）　https://github.com/explodinggradients/ragas

表 2　各モデルについての運用種別の比較についての MRR スコア．列単位でのスコア値となっているため，行単位での比較はできないものとなっている．取りうる値は 0.25 から 1.00.

Scores	CyberAgent LM	CyberAgent LM2	CyberAgent LM2-chat	ELYZA	ELYZA instruct	StableLM	StableLM instruct
Baseline	0.318	0.304	0.420	0.260	0.425	0.301	0.344
RAG	0.463	0.471	0.629	0.394	**0.708**	0.552	0.667
LoRA	0.447	0.469	0.400	0.626	0.340	0.443	0.368
RAG+ LoRA	**0.651**	**0.836**	**0.633**	**0.798**	0.608	**0.783**	**0.703**

なっている場合が多く，質問に対する回答としては不十分であると考えられるものとなっていた．このような点においても，LLM による絶対的な評価の難しさが表層化されたと考えられる．

モデル種別についての分析　続いて，モデル種別での分析（各行での比較）を考える．Answer relevancy のスコアを見ると，GPT-4 や GPT-3.5-turbo が比較的高いスコアとなっていることが分かる．またこれに続く形で RAG や RAG+LoRA の設定下における chat や instruction 形式での学習が施されたモデルが高いスコアとなっている．一方で BERTScore の F1 値を見ると，CyberAgentLM2 や ELYZA, StableLM などの CLM のみでの学習が施されたモデルでスコアが高くなる傾向となった．しかしながらこれらのモデルの実際の出力は，与えられた関連文書の内容を繰り返し出力するなど，FAQ タスクでのユーザへの回答としては定性的に見て適切ではないと考えられる挙動が散見された．

上述のように，実運用を踏まえた上でのモデル毎の絶対的なスコア算出による性能比較には一定の難しさが伴うことが分かる．そこでこれらの出力品質を GPT-4 に比較させることによって，より直接的な比較を試みる[6]．まずはどの運用形態が優れているかを議論するため，各モデルごとに Baseline，RAG，LoRA，RAG+LoRA を候補に取り比較を行った．その結果を表 2 に示す．結果より，ほとんどのモデルにおいて RAG+LoRA でのスコアが最大となった．また，chat や instruction 形式で学習したモデルについては，RAG と RAG+LoRA の MRR スコアが比較的近くなっていることが分かる．ただし，評価者を GPT-4 としているため，LoRA での学習によって self-enhancement bias [13] が助長された可能性は否めないため，正当な比較である保証は難しいと推察される．よって，これらのモデルに対する fine-tuning の有用性の定量的な測定は厳密には難しく，実際の出力を比較することによる定性的な評価が肝要だと考えられる．

表 3　直接比較評価による MRR スコア（上段）と順位分布（下段）.MRR スコアの取りうる値は 0.333 から 1.00.

Model	MRR score & Rank distribution
CyberAgent LM2-chat	0.588 1: 234, 2: 89, 3: 347
ELYZA instruct	0.621 1: 219, 2: 281, 3: 170
StableLM instruct	0.623 1: 217, 2: 300, 3: 153

最後にオープンソースのモデルを候補に取り，その性能差を明らかにするための直接比較を試みる．具体的には表 1，2 より比較的性能が高いと考えられる CyberAgentLM2-chat(RAG+LoRA)，ELYZA-instruct(RAG)，StableLM-instruct(RAG+LoRA) の直接比較を実施し，その結果を表 3 に示す．ここではより詳細な比較のため，各モデルの出力が何位に順位付けされたかの回数を示す順位分布（下段）も併せて掲載している．これは GPT-4 による比較評価において，各順位に何度選択されたかを示すものとなっている．結果より，いずれのモデルについても上位 1 位に選択された回数がほぼ同数であることが明らかとなった．

5　おわりに

本稿では FAQ タスクにおける RAG の運用という設定において，LLM の fine-tuning の有用性を調査した．また実運用を踏まえた LLM 性能評価についても取り組んだ．結果より chat や instruction 形式で学習されたモデルの性能が全般的に高く，これらのモデル間においては明確な性能差やチューニングの有用性を有意に認めることが難しいものとなった．一方で今回試した全ての評価指標において，FAQ タスクで最も肝要であると考えられる「ユーザーの質問を解決する回答をどれ程生成できていたか」を厳密には測定できておらず，今後はこの点についての評価を方針立てることが重要だと考察される．

6）　比較評価で用いたプロンプトを Appendix C に掲載

謝辞

本論文の執筆にあたりご協力いただきました，AI shift AI チームの皆様および，本研究の実験環境をご提供いただきました CyberAgent group Infrastructure Unit の皆様に深く御礼申し上げます.

参考文献

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. **CoRR**, Vol. abs/2001.08361, , 2020.

[2] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019**, pp. 2463–2473. Association for Computational Linguistics, 2019.

[3] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020**, pp. 5418–5426. Association for Computational Linguistics, 2020.

[4] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, **Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual**, 2020.

[5] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. **CoRR**, Vol. abs/2302.00083, , 2023.

[6] Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. Retrieval augmentation reduces hallucination in conversation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021**, pp. 3784–3803. Association for Computational Linguistics, 2021.

[7] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, **International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA**, Vol. 162 of **Proceedings of Machine Learning Research**, pp. 2206–2240. PMLR, 2022.

[8] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In **The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022**. OpenReview.net, 2022.

[9] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021**, pp. 3045–3059. Association for Computational Linguistics, 2021.

[10] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: retrieval-augmented black-box language models. **CoRR**, Vol. abs/2301.12652, , 2023.

[11] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Scott Yih. RA-DIT: retrieval-augmented dual instruction tuning. **CoRR**, Vol. abs/2310.01352, , 2023.

[12] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In **8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020**. OpenReview.net, 2020.

[13] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. **CoRR**, Vol. abs/2306.05685, , 2023.

A　実験にて使用したモデル一覧

表 4 実験にて使用するモデル一覧．ELYZA および ELYZA-instruct はそれぞれ HuggingFace にて ELYZA-japanese-Llama-2-7b-fast，ELYZA-japanese-Llama-2-7b-fast-instruct として提供されるモデルを指すものとする．また StableLM，StableLM-instruct についても同様に，japanese-stablelm-base-ja_vocab-beta-7b，japanese-stablelm-base-ja_vocab-instruct-7b として提供されるものを指す．

モデル	機構	学習
CyberAgentLM	GPT	Pre-training
CyberAgentLM2	Llama2	Pre-training
CyberAgentLM2-chat	Llama2	Pre-training Chat
ELYZA	Llama2	Pre-training
ELYZA-instruct	Llama2	Pre-training SFT(instruction)
StableLM	GPT	Pre-training
StableLM-instruct	GPT	Pre-training SFT(instruction)

B　各種ハイパーパラメータ設定

表 5 学習時に使用するハイパーパラメータ

Hyper parameters	
Learning rate	5e-5
Quantization	8-bit
Optimizer	Adam 8-bit
LoRA rank	256
LoRA α	516

表 6 推論時におけるテキスト生成に関するハイパーパラメータ

Hyper parameters	
Quantization	8-bit
Beam size	5
Temperature	1.0
Max generate tokens	1024
Early stopping	True

C　比較評価で使用したプロンプト

表 2 および表 3 の評価にて，GPT-4 に与えたプロンプトを掲載する．以下は候補数が 3 つの場合のものである．

タスク定義
ある質問とそれに関連する文書が与えられているとします。これらを踏まえて生成された回答を 3 つ入力するので、回答の質が高い順に並べてください。
答え方としては json 形式で {'index1': {'rank': x, 'reason': foo}, 'index2': {'rank': x, 'reason': foo}, 'index3': {'rank': x, 'reason': foo}} のように、候補インデックスに対する順位とその理由を当てはめる形で答えてください。
従って rank に入る数字は 1、2、3 のいずれかです。それ以外の数字は答えないようにしてください。

判断基準
回答の質を判断する基準として以下の指針を参考にしてください。
- 関連文書の内容を踏まえ、質問にきちんと回答する内容になっているか。
- 関連文書の内容を踏まえ、質問に関係のない内容が含まれていないか。
- 文章として明らかに破綻しているものが回答に含まれていないか。
- 同じ内容の文章が繰り返し出力されていないか。

質問
{query}
関連文書
{content}

回答 index: 1
{ 候補 1}

回答 index: 2
{ 候補 2}

回答 index: 3
{ 候補 3}