

エンジニアリング講義



2025/4/10

開発チーム: ベルトン璃亜武

ベルトン 璃亜武 (べるとん りあむ)



早稲田大学 先進理工学部

LLM講義開発チーム所属

■ 活動内容

- RAGを活用した講義質問対応Chat-botの開発
- GENIACプロジェクト コアメンバー
 - 8BのLLM開発における学習チームのマネジメントを担当



アジェンダ

- 開発の動機
- リサーチ
- 実装 / 評価
- 講義への導入
- 今後の展望

アジェンダ

- 開発の動機
- リサーチ
- 実装 / 評価
- 講義への導入
- 今後の展望

「講義中に寄せられる質問をChatBotに対応させたい」

これまでの課題

- 全ての質問に講師が回答することは難しい
 - 講師側の負担（質問の回答に割くことが可能な時間 [講義内・外]）
- 生徒は講義中の限られた時間でのみしか質問できない



ChatBotを導入することで上記を解決できるのではないか??

仮説（解決できそうな課題）

- 「運営側」のメリット
 - 講師の質問対応へかかる負担&コストが減る
- 「生徒側」のメリット
 - 生徒は自身の好きなタイミングで質問でき、即座に回答を得られる
 - 気軽に質問できるため質問のハードルが低い

開発の条件

- 講義の内容に則して回答すること

アジェンダ

- 開発の動機
- リサーチ
- 実装 / 評価
- 講義への導入
- 今後の展望

1. 過去の事例の分析

どのような質問に対応していくのが良いか、
過去開講講義におけるQAデータを解析する。

2024年度LLM講義にて導入予定

⇒ 2023年度LLM講義でのQAデータを分析

- 全7回の講義
- 2023年度のQAデータは全部で891件

分析結果

質問/回答の分類	件数
講義中に取り上げた内容に関する質問	398
「はい」「いいえ」で回答可能な質問	22
講義スライド自体に関する質問	23
演習に関する質問	19
関連研究を訪ねる質問	6
回答なし	390

過去事例の分析により、ChatBot導入の需要が可視化できる。

また、対処すべきターゲットを「講義中に取り上げた内容に関する質問」にフォーカスすれば良いことも分かる。

2. 手法の検討

a. Finetuning

or

b. RAG (Retrieval Augmented Generation)

ChatBot開発にてパツと思ひ浮かぶのが上記二つの手法。

「ドメイン特有の知識に対応させたい」場合に期待できる。

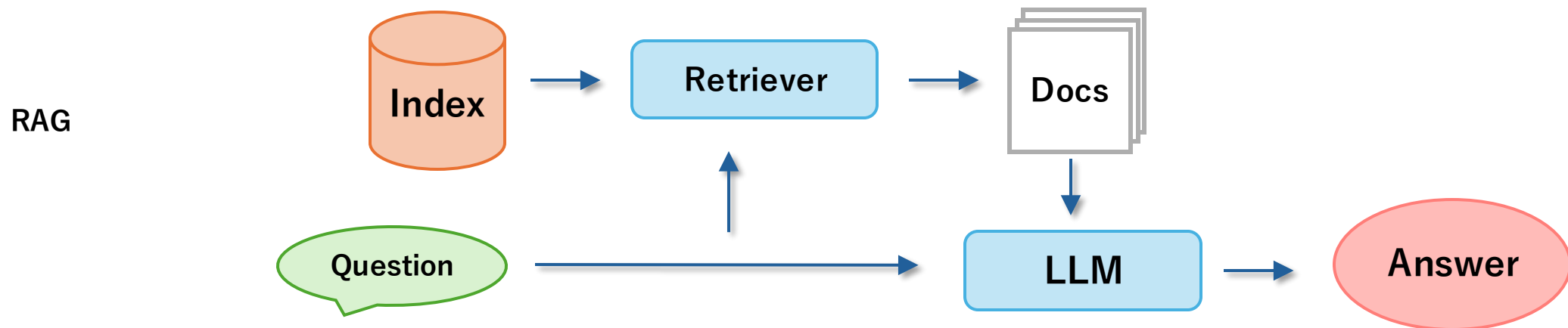
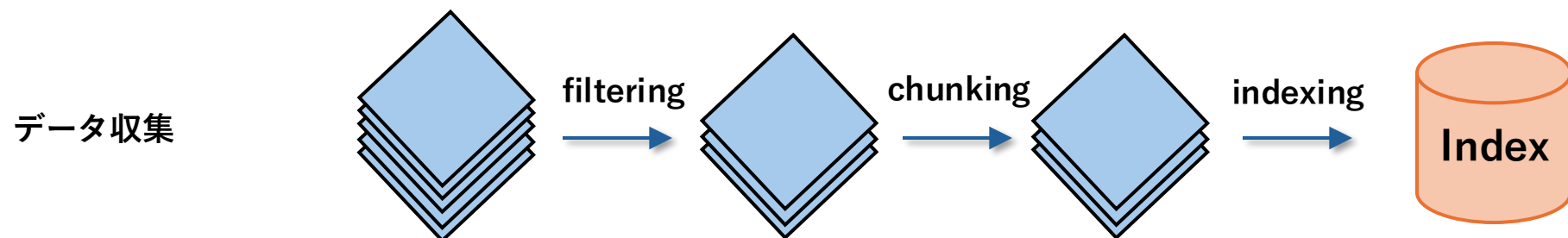
⇒今回は、「講義の内容に則して回答させたい」ので上記手法は有望であった。

補足: RAG

外部の資料（ドキュメント）を回答の根拠として利用する。

正しいドキュメントを取得できるかどうかが鍵

RAGの基本フロー



Finetuning

- Fine-Tuningの目的は??

- 回答に役立つ知識を加えたい

- 既存の質問回答のペアを学習させる / 講義で扱った知識を学習させる



- 学習させるためには手元のデータが不十分

- 以下に示す論文で議論されているが、Fine-Tuningでの知識獲得は微妙

- Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs^[1]
 - Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?^[2]

- そもそも…**講義の内容に則した回答とは??**

→ この辺りが不明瞭だとFineTuningする意義がない。

一方でRAGは...

- 既存のQAデータ(2023 LLM)での検証結果
 - ベースライン (素のGPTモデル)と比較してハルシネーションが激減
 - ⇒ ドメインに有用な知識に対応が可能
 - ⇒ RAGでChatBotを開発するのが良さそう。(この方向で決定)

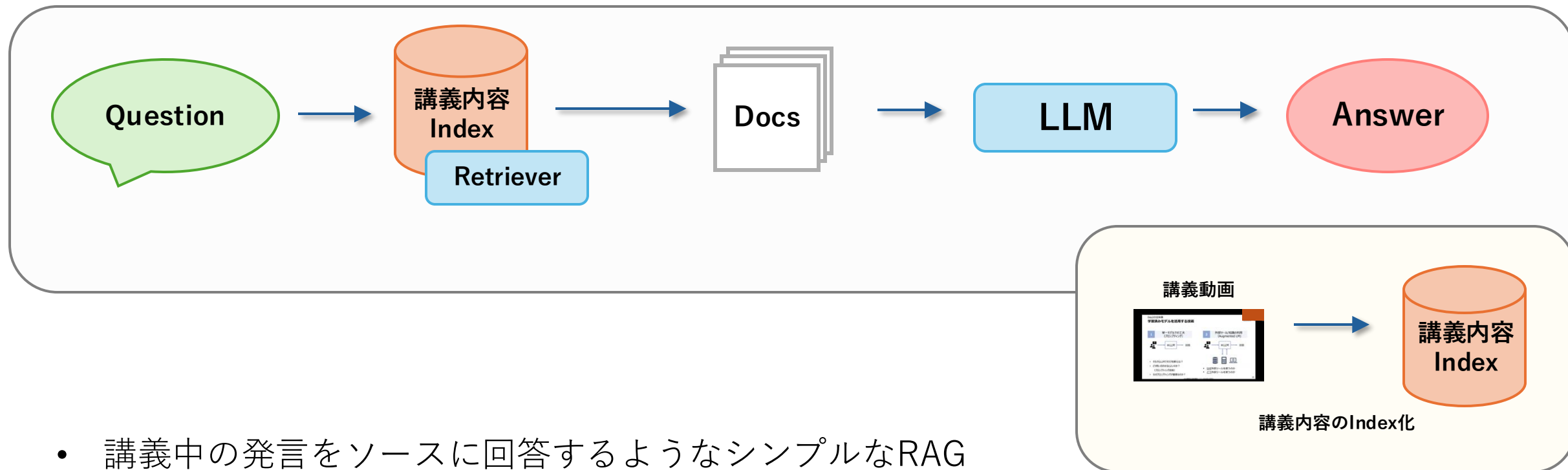
特に..

- RAGは基本的に学習を伴わないで手軽に実装できるのも良い!

アジェンダ

- 開発の動機
- リサーチ
- **実装 / 評価**
- 講義への導入
- 今後の展望

初期の構造



- 講義中の発言をソースに回答するようなシンプルなRAG
 - 今後はこのモデルがベースラインとなって開発していくことになる。
 - では、このモデルはどの程度の性能なのか？
- 評価指標が必要

実装 / 評価

検討した全ての評価

- **生成速度**
- **F1 Score**
- **Mauve**
- **str-em**
- Rouge Score
- FAct Score^[3]
- **Ragas: faithfulness**
- **Ragas: answer relevancy**
- **Ragas: context precision**
- **Ragas: context recall**
- **Ragas: Harmfulness/Toxicity**
- **LlamaIndex: Relevancy**
- **LlamaIndex: faithfulness**
- **DeepEval: Relevancy**
- **DeepEval: faithfulness**
- **DeepEval: Context Relevancy**
- **DeepEval: Bias**
- MRR
- Hit Rate
- FactKB^[4]
- BertScore

- RAGの評価におけるベストプラクティスが当時なかったので、片っ端から試した。


評価の詳細

- 生成速度
質問をしてから回答が得られるまでの速度を測る
- F1 score
正解データとの一致度
- Mauve score
回答の流暢さ（自然な文章かどうか）
- Str-em
回答に正解の単語が含まれているか

実装 / 評価

評価の詳細

- 生成速度
質問をしてから回答が得られるまでの速度を測る

- F1 score
正解データとの一致度 

自由記述形式のQAでは、回答が一意に定まらない



F1 scoreやstr-emでの評価は適切かどうか怪しい

- Mauve score
回答の流暢さ（自然な文章かどうか）

そもそも、正解データを用意するのは
難しかったりもする

- Str-em
回答に正解の単語が含まれているか 

評価の詳細

- Ragas / LlamaIndex Eval / DeepEval / BertScore
 - Faithfulness: ドキュメントに忠実かどうか
 - Answer Relevancy: 生成された回答と質問の関連度
 - Content Precision: Topkドキュメントに含まれる関連するドキュメントの割合
 - Context recall: 回答をクエリとすることで同じドキュメントを取得できるか
 - Harmlessness: 回答の有害性
 - Bias: 文化や性差によるバイアスを含んでいないかどうか

これらはLLM as judgeによる評価。
評価の信頼性の観点から複数の手法(prompt)を採用して各観点の評価に活用

改善.1

何をソース(ドキュメント)としてRAGを構築するのが良いか

考えられるもの

- 「講義中の発言」
zoomで開講された講義での講師の発言を書き起こしたデータ
- 「講義資料」
講義で使われたスライド (パワーポイント)の資料データ
- 「参考文献」
講義で扱われた参考文献となる論文などのデータ
- 「外部検索」
質問をwebブラウザ上で検索した際にヒットするwebページ上の情報

実装 / 評価

改善.1：何をソースにするか

- 「講義中の発言」
ベースライン。これと比較して検証していく。
- 「講義中の発言」の要約をソースにする。
✗性能 ↘
- 「講義中の発言」 + 「参考文献(論文)」
○性能 ↗
- 「講義中の発言」 + 「参考文献(論文)」 + 「外部検索」
✗生成速度 + 「講義に即する」を満たさない恐れ

→ 「講義中の発言」 + 「参考文献」が良さそう

(「講義資料」は図表が中心で、ドキュメントとするには情報が少ないので不採用)

実装 / 評価

改善2：モデルの構造

様々な研究で提案されていた手法での実験

- **In-Context Retrieval Augmented Model**^[5]

✗ 論文通りの結果にならず、評価以前に断念

- **ActiveRAG**^[6]

✗ かなりのハルシネーションが散見され評価以前に断念

- **Self-RAG**^[7]

△性能は良いが、日本語用のモデルがない。(Fine-Tuningするにもデータはどうか)

✗ 生成速度

- **IRCoT**^[8]

✗ コスト

- **Adaptive RAG**^[9]

✗ よく訓練された分類機が必要

→ 様々な研究手法を試したが、実運用する上では課題感が残る手法が多く、採用しなかった。

改善.2：モデルの構造

アイデアベースでのシンプルな改善

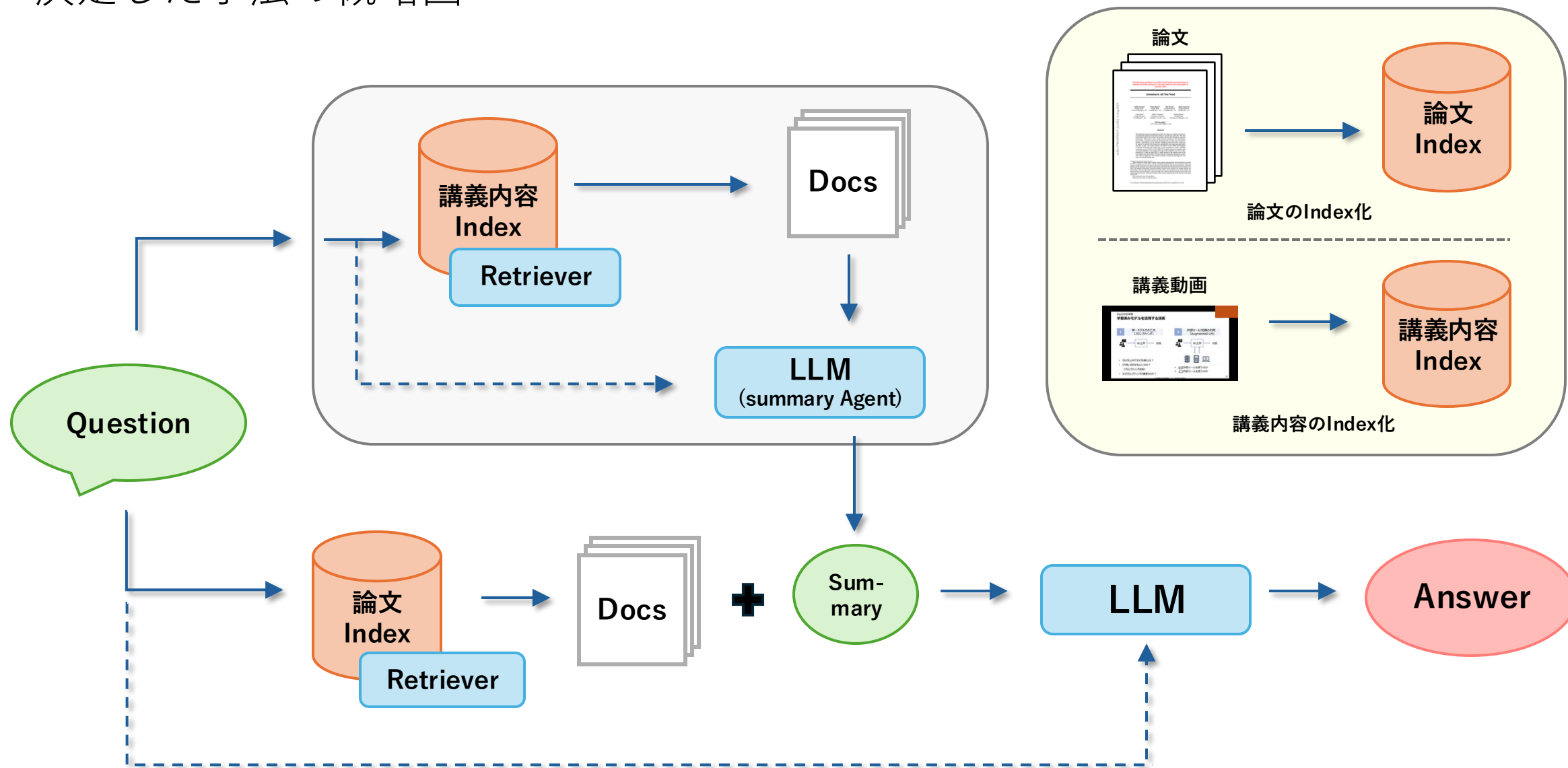
- 1) 「講義中の発言」と「参考文献」を同列に扱う。
- 2) 「講義中の発言」のみでの回答を会話履歴とし、その後「参考文献」を使って再回答する。
- 3) 「講義中の発言」で要約作成。それを踏まえて「参考文献」を使った再回答をさせる。
- 4) 「講義中の発言」「参考文献」それぞれで回答させ、最後に回答をマージする。

タイプ	mauve	f1	str-ex(hit)	ans_relevancy	faithfulness	context_precision	context_recall	harmfulness	bert_precision	bert_recall	bert_f1	invalid
手法(1)	48.1	0.0	38.5	0.77391	0.764801	0.809457	0.529825	0.0	0.638528	0.655672	0.646837	11 / 50
手法(2)	95.7	0.0	25.0	0.847681	0.5625	0.812405	0.61088	0.0	0.629259	0.651741	0.640241	26 / 50
手法(3)	50.4	0.017	34.2	0.871377	0.89812	0.84068	0.54881	0.0	0.741223	0.728849	0.734713	12/50
手法(4)	32.9	0.0	38.8	0.825475	0.837153	0.872676	0.547526	0.0	0.756438	0.734558	0.745005	1/50

→ 上記の他にもコストなどの観点から(3)の手法を採用することに決定した

実装 / 評価

決定した手法の概略図



改善.3 ドキュメントの質

モデルの構造は良く改良できたが、それだけでは出力が不安定

⇒RAGのドキュメントの質は十分か??

実は、単純に「論文をPDF OCRで取得する」「講義中の発言をspeech2textで書き起こす」だけでは文構造や単語に間違いが多く含まれたものになりがちであることがわかった。

- 論文のPDF OCRでの取得
 - pdfのページ間遷移や図表の文中への挿入により、文構造がごちゃごちゃになる
- Speech2textでの講義書き起こし
 - 講師の言い間違いが反映されていなかったり、日英の混合する発言、特定の固有名詞等が誤って書き起こされる

→ 質の良いドキュメントを得るためにはどうすれば良いだろうか？

改善.3 ドキュメントの質

論文データの取得: (この辺りは人目で判別がつくので、人間による評価やコスト的な面で評価)

- PDF OCR

✗ 論文に関して、文構造が崩れる場合が多発。

- TeX

✗ 論文を取り込む + 解析するプログラムの作成の手間

- HTML化

○ 文構造を壊さず、簡単にテキストを読み込むことができる。

→ 高品質なドキュメントの取得が可能になった

改善.3 ドキュメントの質

書き起こしデータの修正:

- text2speech
 - ✗ 不自然な日本語データになる。
- LLMによる修正
 - ✗ 講義資料と誤った書き起こしデータを渡して、修正させるプロンプトだけでは、ハルシネーションを引き起こすような講義に忠実でないものが得られる

→ 果たして、最適な方法はなんだろうか？（最終課題テーマの一つ）

アジェンダ

- 開発の動機
- リサーチ
- 実装 / 評価
- **講義への導入**
- 今後の展望

講義への導入

- アプリケーションへの導入

実運用上は、これまでのような回答の質だけでなく、UXの向上も求められる。
UX向上のためにはどのような機能があれば良さそうか??

- マルチターンChat形式でのQA機能
- 回答の根拠の参照
- 他の受講生のQA参照
- フィードバック機能
- QA検索機能

→ まずは思いつく機能を盛り込む

講義への導入

- Omnicanpusへの導入
 - 前述した機能をあらかじめ取り入れて実装
 - Omnicanpusでの実装は、「手軽に質問」する上でハードルか？

Omnicanpus

LI

QA

[トップ](#) / [LLM2024](#) / [QA一覧](#)

RAGについて教えてください。

RAG（Retrieval-Augmented Generation）は、関連する文書や情報を取得し、その情報に基づいて言語モデルが応答を生成する手法です。この方法は、外部の知識を取り入れることで生成されるテキストの質と関連性を向上させるために使用されます。RAGは、テキスト、画像、ビデオなどさまざまなコンテキストで適用可能であり、文脈に即した正確な出力を生成するための多用途なツールです。このプロセスは通常、関連する文書を検索し、それらを入力クエリと組み合わせてから、言語モデルを使用して応答を生成することを含みます。このアプローチは、言語モデルのトレーニングデータを越えた外部知識を必要とする複雑なクエリに対処する際に特に有用です。RAGは、外部のデータベースや知識源にアクセスし利用することで、言語モデルの能力を強化する方法と見なされています。[0]

使用した参考文献: GPT-4oの知識をもとに回答しています。

RAGで取得した文書一覧:

- ▶ [\[1\].\[2406.06608\] The Prompt Report: A Systematic Survey of Prompting Techniques](#)
- ▶ [\[2\] A Survey of Large Language Models in Finance \(FinLLMs\)](#)
- ▶ [\[3\] Benchmarking Retrieval-Augmented Generation for Medicine](#)
- ▶ [\[4\] A Human-Inspired Reading Agent with Gist Memory of Very Long Contexts](#)
- ▶ [\[5\] RAGGED: Towards Informed Design of Retrieval Augmented Generation Systems](#)

質問は150文字以内にしてください。講義内容の質問は聞きたい部分を細かに記述してください

送信

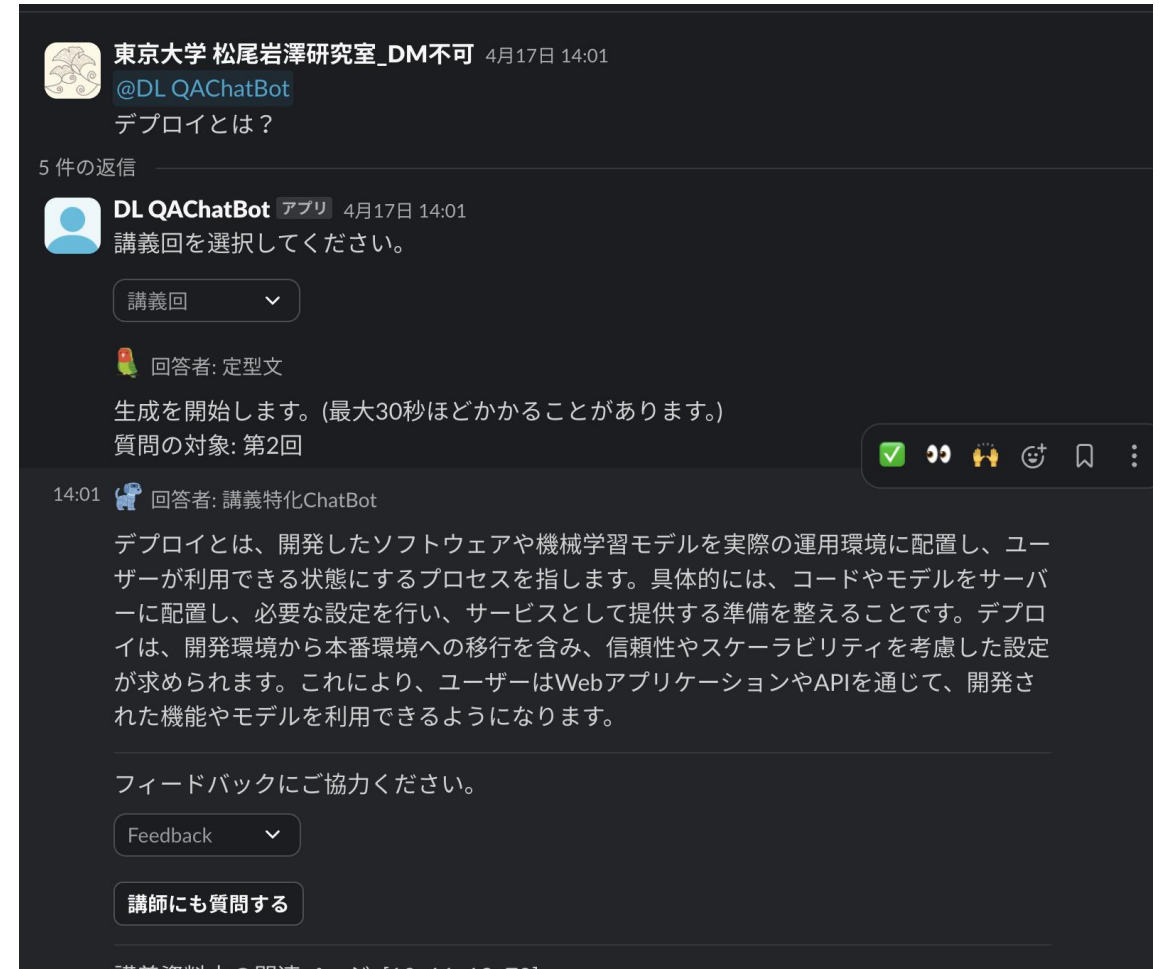
解決しましたか？

*フィードバックを返すと受講者側はこのセッションでは追加の質問ができません。



講義への導入

- Slackへの導入
 - 前述した機能をあらかじめ取り入れて実装
 - 「講師へ直接質問」できるような機能も追加
 - スマートフォンからも気軽に質問できる上に、管理も楽になった。



講義への導入

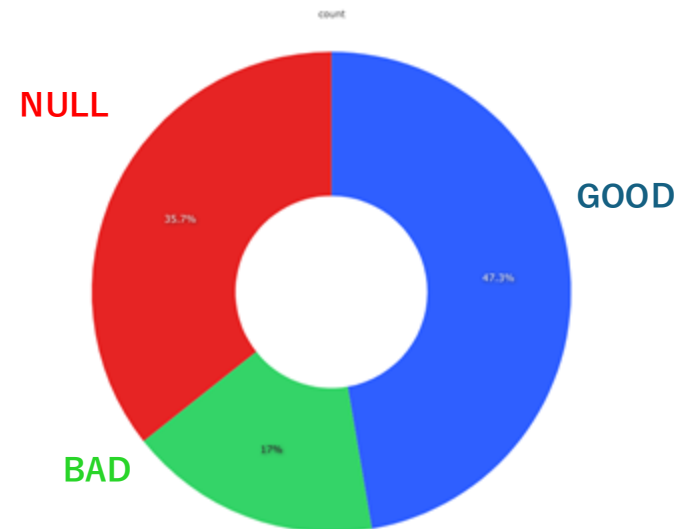
- 講義導入以降の評価
 - 講義導入以降は、実際に活用した受講生からのフィードバックや、使用状況、インタビュー等によって使用感を評価した。
 - インタビューを通して、UXは「Slack > Omnicampus」が良いことなどが分かってくる。

フィードバック

- 受講生からの回答に対するフィードバックを収集した。
- 回答してくれた結果を見ると、過半数以上が満足だったことなどが分かる

使用状況の評価

- 全体で5800件の質問が受講生から寄せられた
 - 2023年度と比較して6倍以上の質問数
- 質問を気軽に行えるようにできたか



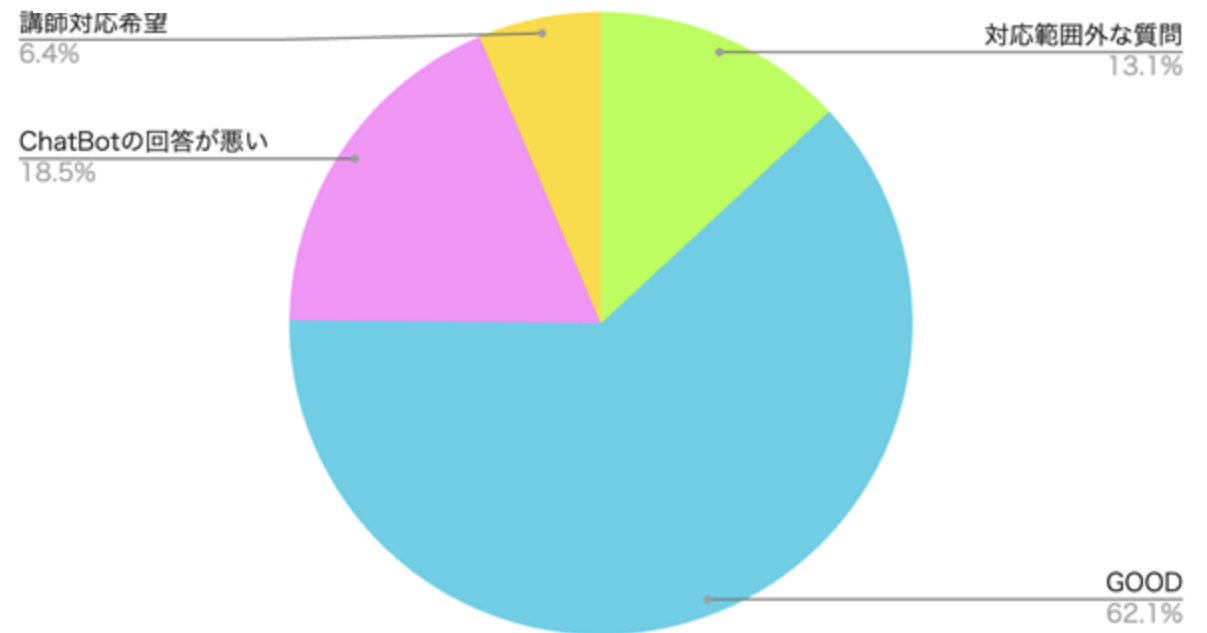
本講義全体での受講生によるFB

講義への導入

- 講義導入以降の評価
 - また、「どのような質問が寄せられたか」や「BadなQAはどのようなものか」の分析も定性的に行った。

講義で寄せられた質問の分類

- 講義で扱った知識の復習を目的とした質問
- 講義で扱った知識等に関して納得や理解を目的とした質問
- 講義で扱った文献に対して更に詳細な知識を求める質問
- 関連する発展的な研究に関して知識を求める質問
- 自身のアイデアに基づく手法の有用性や実現可能性を確かめる質問
- 自身の関心ある研究の論文を挙げさせる質問
- 意見を求める質問
- 運営・業務に関する質問
- 演習などコーディングに関する質問



11/25時点での直近300件の人手評価

→ 質問の傾向は変わらないが、総数が増えたことで今後の対応における課題感などが浮き彫りに

アジェンダ

- 開発の動機
- リサーチ
- 実装 / 評価
- 講義への導入
- 今後の展望

今後の展望

- LLM2024講座での導入で様々な成功例や課題感を獲得

今後の取り組み

- 細かい機能改善
- 他講義への展開（現在DL基礎講座、PhysicalAI講座にも導入）
- 得られたデータを用いたFine-Tuningなどモデルの学習も視野に入る

参考文献

- [1] Ovadia, O., Brief, M., Mishaeli, M., & Elisha, O. (2023) Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs. arXiv. <https://arxiv.org/abs/2312.05934>
- [2] Gekhman, Z., Yona, G., Aharoni, R., Eyal, M., Feder, A., Reichart, R., & Herzig, J. (2024) Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? arXiv. <https://arxiv.org/abs/2405.05904>
- [3] Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W., Koh, P. W., Iyyer, M., Zettlemoyer, L., & Hajishirzi, H. (2023). FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. arXiv. <https://arxiv.org/abs/2305.14251>
- [4] Feng, S., Balachandran, V., Bai, Y., & Tsvetkov, Y. (2023). FactKB: Generalizable Factuality Evaluation using Language Models Enhanced with Factual Knowledge. arXiv. <https://arxiv.org/abs/2305.08281>
- [5] Ram, O., Levine, Y., Dalmedigos, I., Muhlgaay, D., Shashua, A., Leyton-Brown, K., & Shoham, Y. (2023). In- Context Retrieval-Augmented Language Models. arXiv. <https://arxiv.org/abs/2302.00083>
- [6] Xu, Z., Liu, Z., Yan, Y., Wang, S., Yu, S., Zeng, Z., Xiao, C., Liu, Z., Yu, G., & Xiong, C. (2024) ActiveRAG: Autonomously Knowledge Assimilation and Accommodation through Retrieval-Augmented Agents. arXiv. <https://arxiv.org/abs/2402.13547>
- [7] Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. arXiv. <https://arxiv.org/abs/2310.11511>
- [8] Trivedi, H., Balasubramanian, N., Khot, T., & Sabharwal, A. (2022). Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions arXiv <https://arxiv.org/abs/2212.10509>
- [9] Jeong, S., Baek, J., Cho, S., Hwang, S. J., & Park, J. C. (2024) Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity arXiv. <https://arxiv.org/abs/2403.14403>