

RAG-1グランプリ知見レポート

※本資料の無断での転載・再配布はご遠慮ください。



多数の皆様にご参加いただき感謝です！





RAG-1グランプリ
みんなで生成AIの精度限界に挑戦！「ハルシネーション」の壁を越えよう！









 SIGNATE

🕒 締切：2024年10月10日 23時59分59秒 📄 投稿：3634件 👤 参加：1464人

参加者数	投稿チーム数	投稿件数	ベストスコア
1464名	505チーム	3634件	0.967

入賞おめでとうございます！

順位	チーム名 / ユーザ名	暫定評価	最終評価 ▼	投稿 件数
1	50-50 	0.9666667	0.9666667	70
2	gen99 	0.9666667	0.9666667	33
3	monach 	0.8916667	0.8916667	26
4	sugapussy    	0.8916667	0.8916667	40
5	AioRaito 	0.8916667	0.8916667	22
6	KazRico 	0.8833333	0.8833333	10
7	titleist980 	0.8750000	0.8750000	26
8	wisac 	0.8666667	0.8666667	24
9	suzookita 	0.8666667	0.8666667	26
10	HF-1 	0.8583333	0.8583333	19

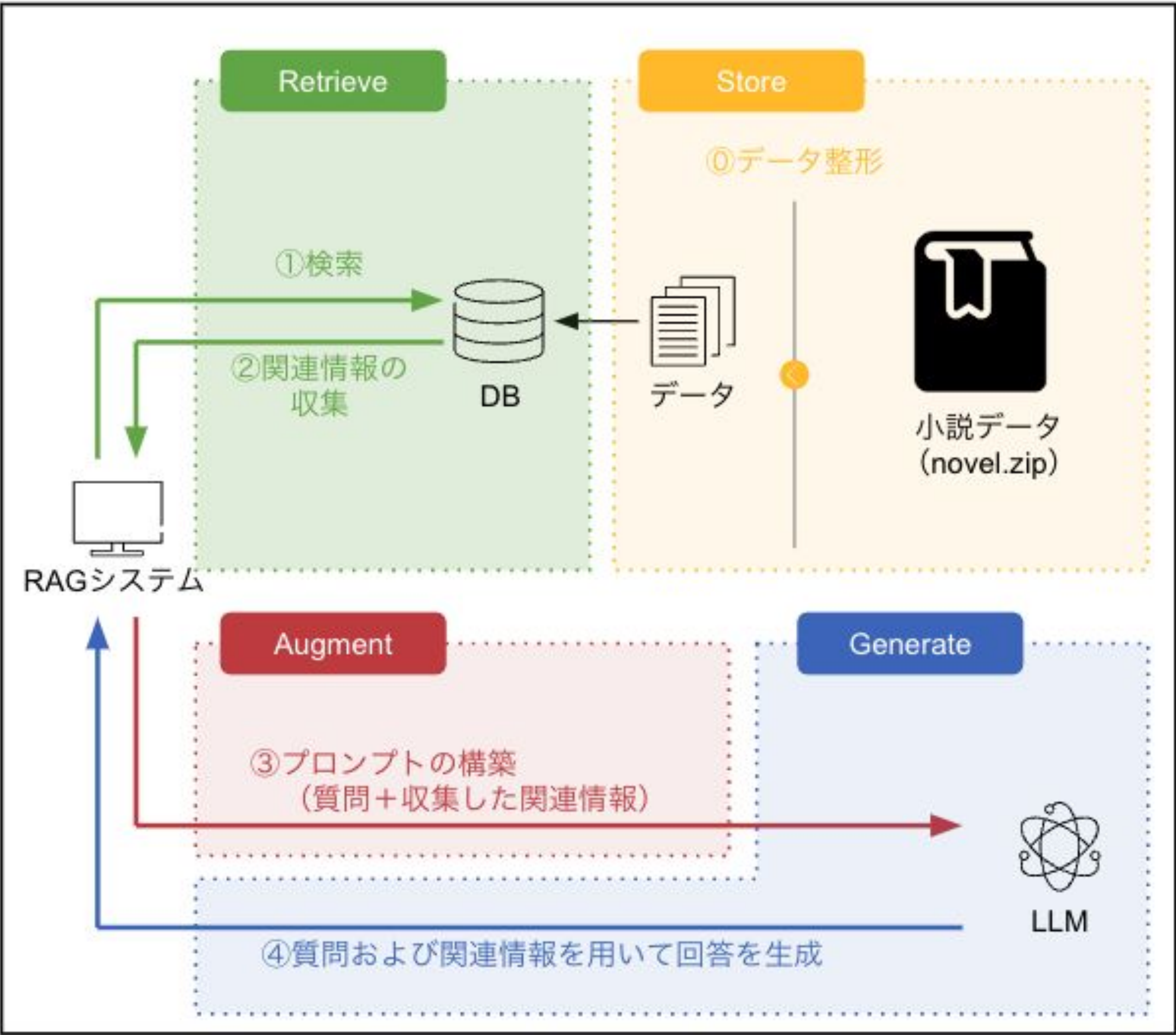
順位	チーム名 / ユーザ名	暫定評価	最終評価 ▼	投稿 件数
11	niyo 	0.8416667	0.8416667	11
12	shunya nagashima 	0.8250000	0.8250000	10
13	yuki_shino 	0.8166667	0.8166667	35
14	shinnosuke hirano 	0.8166667	0.8166667	6
15	yuki5656 	0.8000000	0.8000000	31
16	Team H  	0.8000000	0.8000000	54
17	N1 	0.7916667	0.7916667	24
18	ANTEL 	0.7750000	0.7750000	20
19	kesumete 	0.7416667	0.7416667	31
20	naoki1213mj 	0.7333333	0.7333333	13

小説データから、その小説に関する質問に回答するRAGを構築

質問データ

index	質問
1	主人公誰？
2	登場人物は何人？
...	...

参加者のタスク(本コンペのRAG構成例)

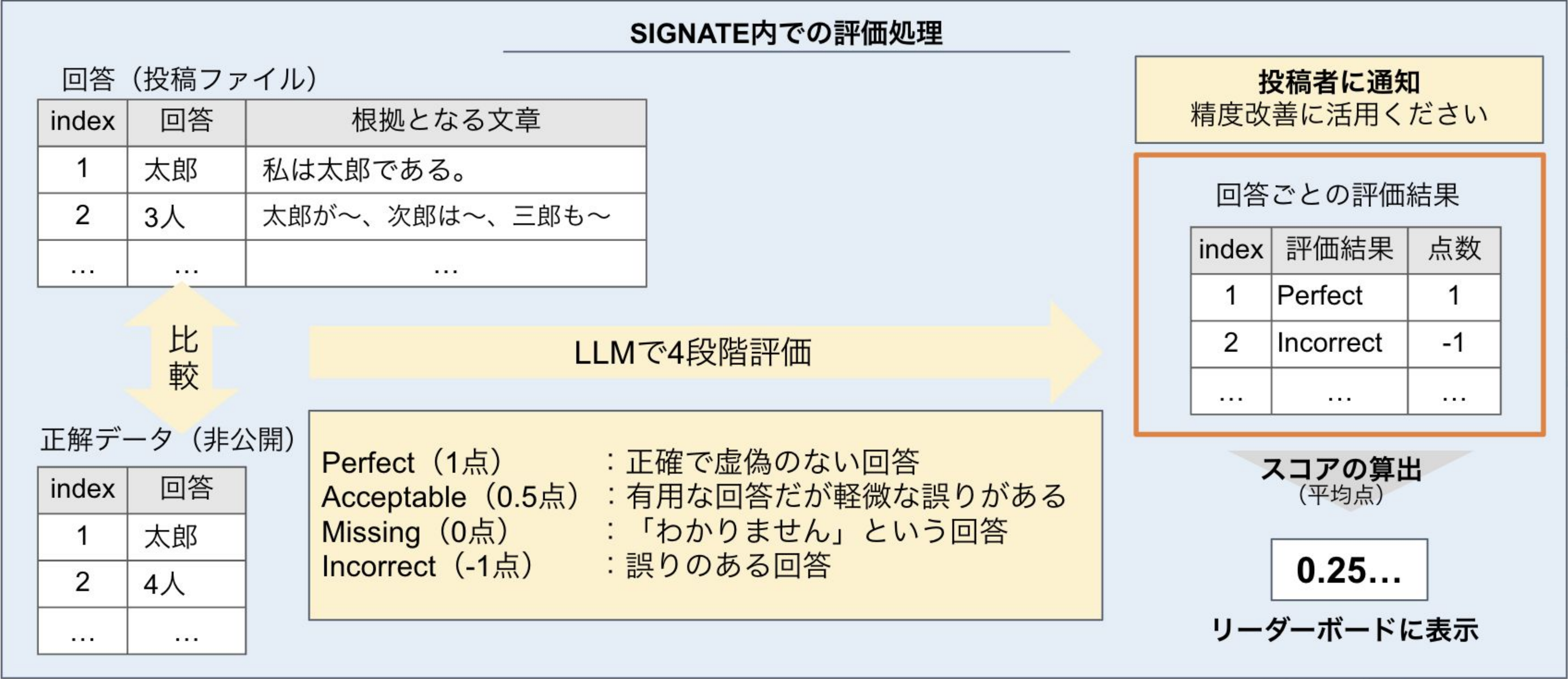


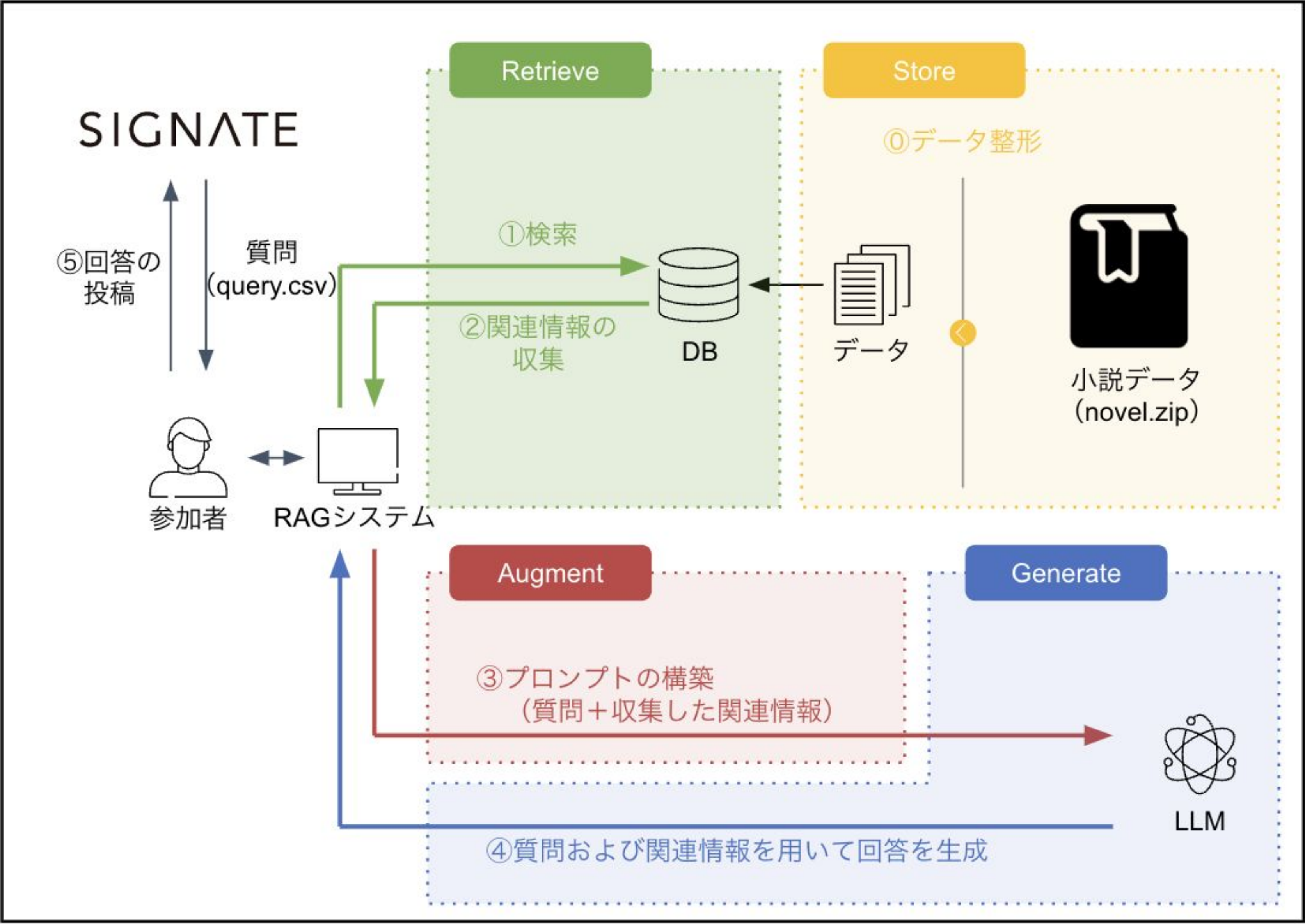
回答(投稿ファイル)

index	回答	根拠となる文章
1	太郎	私は太郎である。
2	3人	太郎が～、次郎は～、三郎も～
...

SIGNATEに投稿

LLMを使って回答ごとに採点し、その平均点をスコアとします





【Store】
参照データの整形格納

【Retrieve】
関連データの取得

【Augment】
回答生成の指示

【Generate】
回答文の生成

アプローチ	解説
事前処理	ルビ、記号、特殊文字削除。文字コード変換など。
抽出処理	形態素解析、固有名詞抽出、タイトル抽出、key-Value抽出。
要約処理	LLMによる要約、マークダウン表記。
分割処理	単語単位・文単位・章単位など様々な粒度でチャンク分割、ベクトル化。
特殊処理	現代語への翻訳。

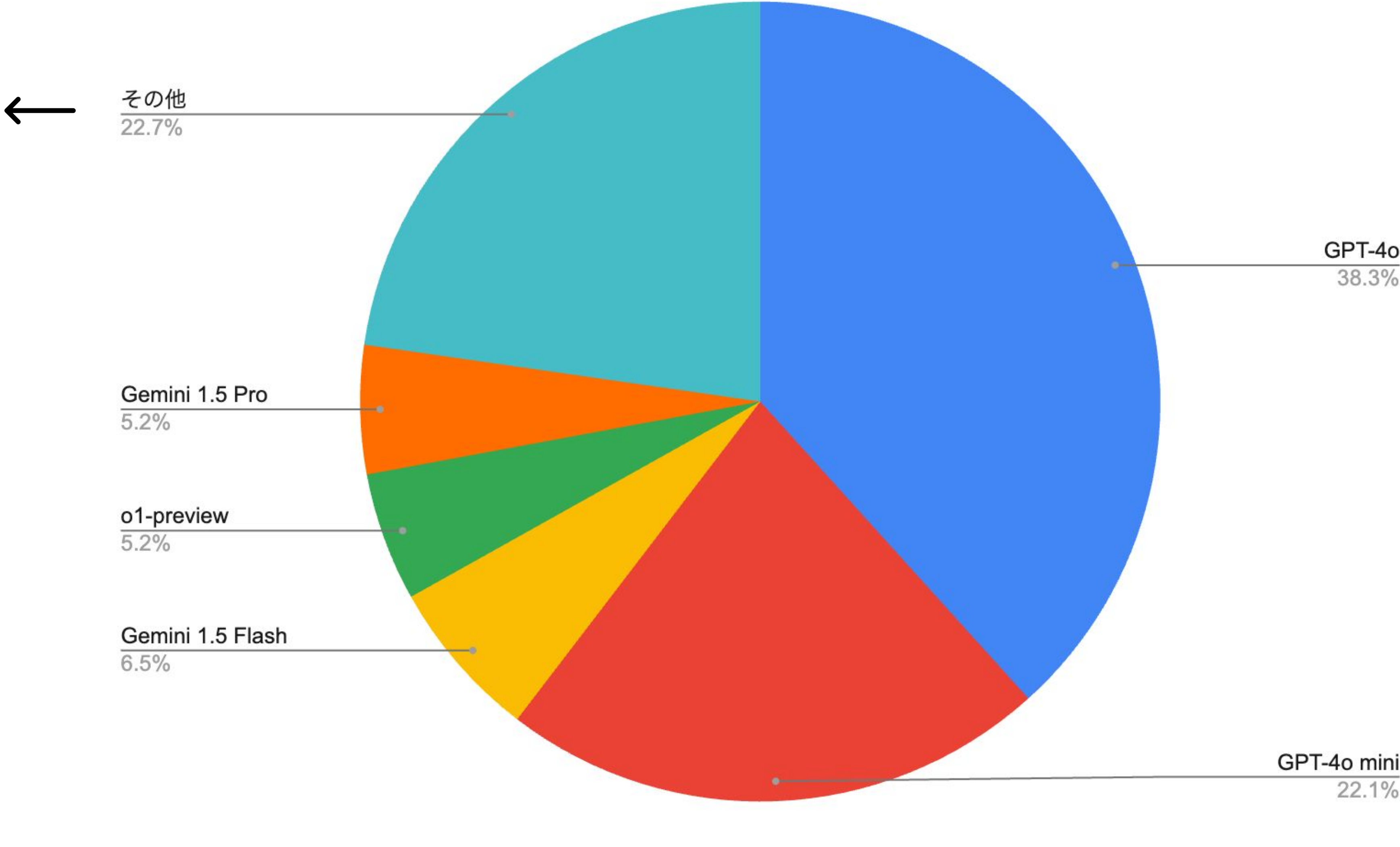
アプローチ	解説
RAG関連システムの利用	GraphRAG、Neo4j、Azure AI Search、Pinecone、AWS Bedrock、Cohereなど。
小説の特定	質問に関連度の高い小説を特定する。
関連情報の特定	タイトル・キーワード・ベクトル検索など。
広域検索性能向上	Map-reduce型など。
検索精度向上	Reranking。複数回検索によるRefine。

アプローチ	解説
プロンプトエンジニアリング	注意事項、文字数制限、フォーマット指示。回答根拠や確信度の出力。 Few/Many-shot learning。
質問の分類	質問のタイプを事前に分類、タイプ別にプロンプトを最適化。
推論機能の実装	生成処理の連鎖など多段階処理の実装。Chain of thought。
アンサンブル	複数のRAGシステム出力の評価・選択。複数のLLMの使用。
参照情報の明確化	タグ、マークダウンにより参照箇所を明示的に出力。

アプローチ	解説
LLMの選定・チューニング	回答生成に使用するLLMの選定。パラメータチューニング。
再考処理	回答基準を満たさない場合、再考させる。回答の品質判定・補正。
外部ツールの自動実行	状況を判断し外部Pythonの実行など。Function Calling。
出力の理解	回答根拠を出力。生成結果の可視化ツール。
ハルシネーション対策	「質問誤り」「わかりません」を生成選択肢にいれる。

使用されたLLMはGPT-4o, miniで約70%を占める。オープンソースLLMや日本語特化LLMなど幅広いモデルの利用もみられた。

生成AIモデル使用人数割合

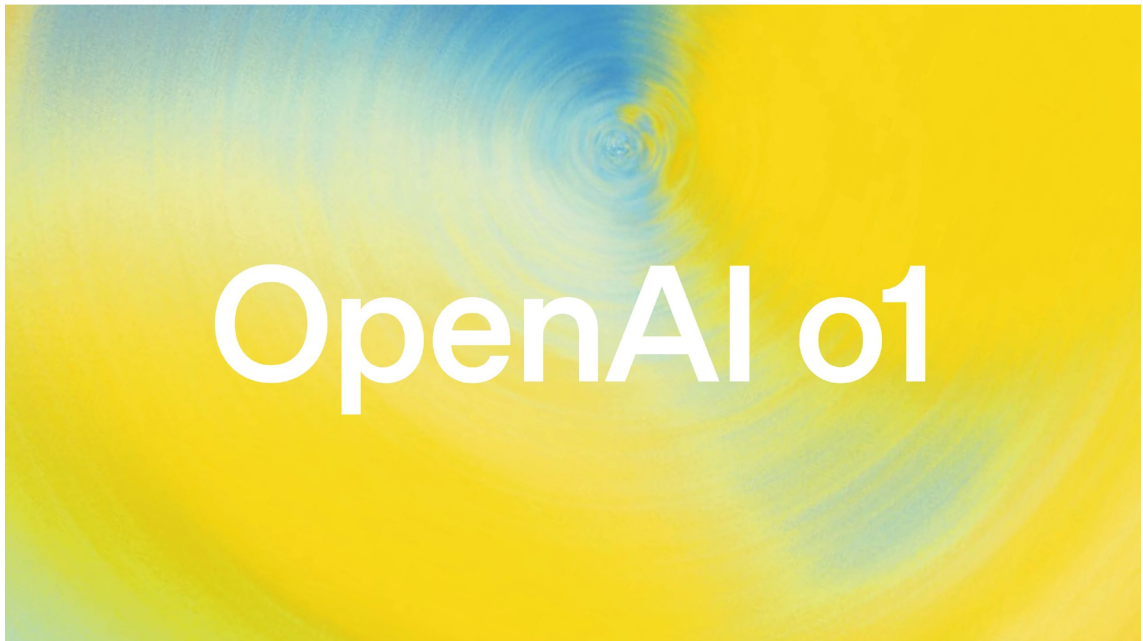


モデル	%
Llama3.2	2.6
gpt-4	1.9
Llama3.1	1.9
Llama3	1.9
Claude 3.5 Sonnet	1.9
gpt-4-turbo	1.3
PLaMo	1.3
-	1.3
Claude 3.5 Haiku	0.6
sentence-transformers	0.6
Dbrx	0.6
Llama-3-ELYZA-JP-8B	0.6
llm-jp-1.3b-v1.0	0.6
Tanuki-8x8B	0.6
text-embedding-3-small	0.6
all-MiniLM-L6-v2	0.6
qwen2.5:72b	0.6
gemma-2-2b-jpn-it-Q8	0.6
Gemma2	0.6
Swallow(llama3.1ベース)	0.6
Quen	0.6

入賞者が利用したLLMはGPTシリーズが支配的であった。

順位	LLM（メイン）	LLM（サブ）	その他モデル
1	o1-preview		
2	GPT-4o		text-embeddings-3-large
3	GPT（詳細記載なし）		
4	GPT-4o		
5	GPT-4o		text-embedding-3-large
6	o1-preview	GPT-4o	text-embedding-ada-002
7	Gemini-1.5-flash-002		
8	GPT-4o		text-embedding-ada-002
9	GPT-4o		
10	GPT-4o	Gemini-1.5-flash-002	
11	Claude 3.5 Sonnet	GPT-4o	text-embedding-3-large
12	o1-preview	GPT-4o	
13	GPT-4o	GPT-4o mini	rerank-multilingual-v3.0（cohere）
14	o1-preview		
15	GPT-4o		rerank-multilingual-v3.0（cohere）
16	GPT-4o		multilingual-e5-large
17	GPT-4o	GPT-4o mini	text-embedding-3-large
18	GPT-4o	Gemini 1.5 Pro	text-embedding-3-small
19	GPT（詳細記載なし）		
20	gemini-1.5-pro-002		

- ・モデルはGPT-4oが最多
最新のo1-previewもちらほら
- ・複数のLLMを組み合わせる
アプローチも
- ・関連文章の取得にエンベディング
やリランク技術も利用されている
- ・ベクトル検索は必須ではない



2024/9/12 開会式のタイミングでリリース
人间的に問題を深く考察するように訓練されている。
従来より高度な問題解決が可能。
博士号を持つ専門家と同等の推論能力を発揮。

設定された60の質問は8つのカテゴリに属する。小説がテーマではあるが、ビジネスケースへの対応を意識して設計されている。

No	質問 カテゴリ	定義	今回の問題における質問例	ビジネスでの対応例
1	単純質問	時間の経過によって 変わることがない 事実を尋ねる質問	作中に登場する通訳者の名前は？	「会社の設立日はいつ？」
2	比較質問	2つの実体を比較する質問	骸骨男にさらわれたのは正一君とミヨ子ちゃんの どちらですか？	「今期と昨期で第1四半期の売上はどちらが 高い？」
3	条件付き 質問	ある条件を指定して 単純な事実を尋ねる質問	仁右衛門が村に到着したときに持っていた家畜は 何ですか？	「赤坂支店の住所を教えて」
4	集計質問	情報の集計を必要とする質問	文中での「大豆」という文字の登場回数は何回 ですか？	「過去5年のXプロジェクトの売上合計は？」
5	誤った前提の 質問	誤った前提や仮定を持つ質問	お房が小諸を出発した日に一緒に出かけた友達 は誰でしたか？（実際には一人で出かけた）	ユーザーが誤った質問（対応する事実が存在しない） をした場合、AIが正しく質問の誤りを指摘できるか？
6	多段階質問	複数の情報を連鎖させて 答えを導く必要がある質問	染物屋の家族が、泥棒から身を守ることができた 理由は何ですか？ （複数の記載を連鎖して考えると理由がわかる）	「今期のキャンペーンでROIが最も高いものは？ そのキャンペーンのチャネルは何？」
7	後処理が 必要な質問	情報を取得した後に 推論や処理を必要とする質問	作中の人力車夫の弟が、1ヶ月（30日）働いた ときの人力車のレンタル代はいくらですか？	「Xサービスを3ヶ月使うとコストはいくらに なりますか？」
8	集合質問	答えとして一連の 実体や物を期待する質問	小説「芽生」で、主人公が借りることができた 庭に植えられていたものとして描写している 植物を全てあげてください。（読点区切り）	「資料に書かれているすべての人名を教えて」

推論や後処理、列挙タスクを含む質問が比較的難易度が高いことがあきらかに。

上位20名の質問カテゴリ別の平均スコア分布

		Tier1		Tier2								Tier3							Tier4			平均
No	質問カテゴリ 順位	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
1	単純質問	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00	0.94	0.83	1.00	0.67	0.89	0.78	1.00	0.94	0.94	0.94	0.94
2	比較質問	1.00	1.00	1.00	0.86	1.00	1.00	0.86	1.00	0.71	1.00	1.00	0.93	0.71	0.93	0.86	1.00	1.00	0.93	1.00	0.71	0.93
3	条件付き質問	0.80	1.00	0.90	1.00	1.00	0.50	1.00	1.00	1.00	0.90	1.00	0.90	1.00	0.80	0.70	1.00	0.60	0.90	0.80	0.60	0.87
4	集計質問	1.00	1.00	1.00	1.00	0.86	1.00	1.00	1.00	1.00	0.71	0.71	0.86	0.29	0.86	0.71	0.93	1.00	0.71	0.29	1.00	0.85
5	誤った前提のある質問	1.00	1.00	0.75	1.00	1.00	0.88	0.75	0.75	0.63	1.00	1.00	0.88	1.00	0.88	0.88	0.75	0.75	0.50	0.88	0.50	0.84
6	多段階質問	0.94	0.83	0.83	0.83	0.83	0.89	0.83	0.83	0.83	0.50	0.83	0.67	0.89	0.83	0.83	0.72	0.72	0.78	0.78	0.67	0.79
7	後処理が必要な質問	0.94	1.00	0.75	0.81	0.69	0.94	0.69	0.63	0.94	0.94	0.50	0.88	0.88	0.88	0.69	0.69	0.63	0.69	0.50	0.88	0.78
8	集合質問	1.00	0.93	0.93	0.64	0.79	0.71	0.93	0.79	0.93	0.86	0.79	0.71	0.71	0.71	0.79	0.64	0.57	0.79	0.71	0.50	0.77
	平均	0.96	0.97	0.90	0.89	0.90	0.86	0.88	0.87	0.87	0.86	0.85	0.83	0.81	0.82	0.79	0.81	0.78	0.78	0.74	0.73	



カテゴリー5

- ・変な質問に正しく「間違いです」といえる能力は重要
（※ビジネスでのハルシネーション対策としても）



カテゴリー6・7・8

- ・情報取得の正確性が前提の上推論性能で差がつく
- ・RAGのタスクでは単純な数え上げは意外に難しい

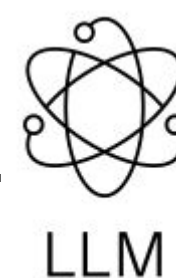
質問カテゴリ7.後処理が必要な質問 - 具体例

項目	内容
質問	馬に乗った骸骨男を見つけてから、明智が投げ縄をするまでに縮めた差は何メートルだと考えられますか？
正解	90メートル
誤答例	「わかりません」「40メートル」「30メートル」
平均スコア (上位20位)	-0.03
小説記述	<p>・“骸骨男のウマは、それよりもっと先を走っているらしいのですが、夜のことでですから、はっきり見えません。警察自動車には、小型のサーチライトがつみこんでありました。ひとりの警官が、それをとり出して、自動車を走らせたまま、屋根の前にとりつけたスイッチをおしますと、パアッと光の棒がのびて、百メートルも先を白く照らしだしました。「アッ、ずっと向こうを骸骨男のウマが走っている。……アッ、角をまがったぞ。よし、うしろの車に近道をして、あいつの前に出るように通信したまえ。」”</p> <p>から、発見した時は100メートルの差があることがわかる。</p> <p>・“骸骨男とのあいだが、だんだん、ちぢまっていきました。四十メートル、三十メートル、二十メートル、ああ、もう十メートルほどになりました。手に汗にぎる競馬です。うしろのウマが、ぐんぐん、前のウマにせまっているのです。そのとき明智は、たずなをはなして、腰のかげんでウマを走らせながら、両手ではそびきのたばをほぐし、結むすび玉だまをつくって、大きな輪にしました。そして、それを右手に持って、クルッと頭の上でまわしはじめたのです。”</p> <p>から、10メートルまで縮めて投げ縄を使った。</p> <p>上記から、100メートルから10メートルまで90メートルの距離を縮めたと考えられる。</p>

①【Store】参照データの整形格納 フェーズの工夫点

- LLMによる数値情報を省略しない小説の要約

そのばん、八時ごろのことです。笠原サーカス団長のふたりのかわいい子どもがおとうさんの団長の帰ってくるのを待っていました。
にいさんは笠原 | 正一《しょういち》といって、小学校六年生、妹はミヨ子といって、小学校三年生です。...

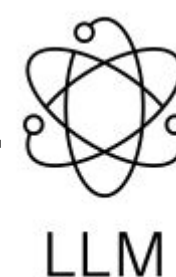


LLM

ある晩の**午後8時**ごろ、サーカス団長の子どもである**小学6年生**の正一君と**小学3年生**のミヨ子ちゃんは、父親の帰りを待っていました。...

- LLMによる主語・目的語が明確になるような質問文の言い換え

「馬に乗った骸骨男を見つけてから、明智が投げ縄をするまでに縮めた差は何メートルだと考えられますか？」



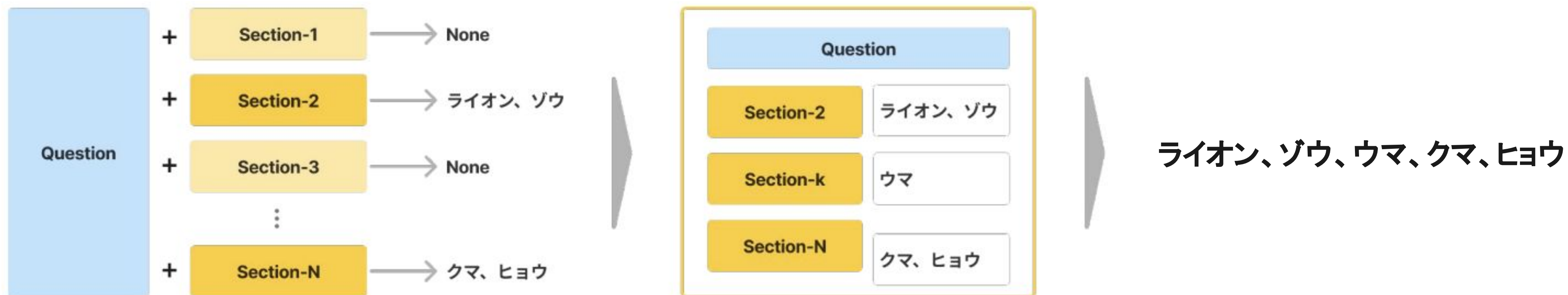
LLM

「馬に乗った骸骨男を**最初に**見つけてから、明智が投げ縄をするまでに縮めた差は何メートルだと考えられますか？」

項目	内容
質問	文中に登場するカタカナ表記された北海道の地名を全て挙げてください。
正解	マッカリヌプリ、シリベシ、ニセコアン、ルベシベ
誤答例	「北海道、マッカリヌプリ、胆振、内浦湾、昆布岳、市街地、倶知安、ニセコアン、函館、ルベシベ」
平均スコア (上位20位)	0.65
小説記述	<div><div><div>・ 蝦夷富士《えぞふじ》といわれるマッカリヌプリの麓《ふもと》</div><div>・ シリベシ河のかすかな水の音だけが聞こえていた。</div><div>・ ニセコアンにしろ、ルベシベにしろ、みなこのような風景が見られる。</div></div><div>上記から、「マッカリヌプリ、シリベシ、ニセコアン、ルベシベ」が相当する。</div></div>

②【Retrieve】関連データの取得 フェーズの工夫点

- Map-Reduce
 - データを小さな部分 (Mapフェーズ) に分割して個別に処理し、その結果を集約 (Reduceフェーズ) して最終的な結果を得る



Q

確信度を有効に使うために工夫されたことはありますか？

回答の確信度は、その数値が低い場合に別のアプローチを行うなどの活用方法が考えられますが、今回は確信度そのものを効果的に使うのは難しかったです。一方で、その確信度の根拠となる文章を参照することで、システム全体の精度を高めたり、その後の処理に役立てたりすることができました。

Q

RAGの開発プロセスをどのように確認・検証したか教えてください。

Weights & BiasesのWeaveというツールを使って、開発したRAGが生成する回答を一つずつ確認しました。

Q

どのような Graph構造を意識して作成しましたか？

LangChainのLLM Graph Transformerを使用した後、LangChainのCypherChainを使用していました。CypherChainは質問に対するCypherコマンドを自動で生成してくれますが、これを利用するとノード間の関連性やプロパティが不十分であることが判明しました。そこで、LLMを用いて関連性を詳細に抽出する独自のChainを作成し、それらをGraphDBに追加する処理を実施しました。

Q

特に有効だったデータ整形の手法を教えてください。

LLMを使って質問に直結するような要約処理が有効でした。今回のコンペでいうと「人の名前」「動物名」「数字」「地名」に注目した要約が有効だったと思います。

一方で、登場人物がかなり多い場合に「人の名前」で要約しても、回答精度は上げるのは難しかったです。

Q

チャンキングの手法について、実装例を教えてください。

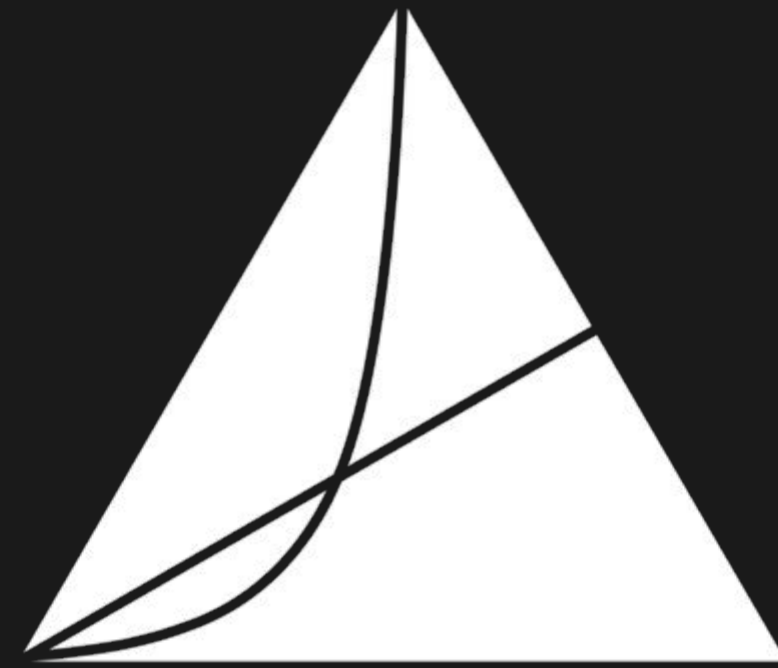
Graphデータベースでの実装例になります。チャンキングを細かくするほど、文章中のエンティティ(重要な要素や概念)を多く抽出することができます。したがって、検索がうまく機能しない小説に対しては、より細かくチャンキングすることで対応しました。

【総括】

- ・今回の課題は、皆様からの提案システムにより、ほぼ解決されたという認識
 - ・設問は8つのカテゴリーに対応し「推論・列挙」問題が比較的難易度が高い
 - ・適切な前処理・推論・後処理により実用精度のRAG構築は可能という所感
-
- ・このあと、上位者のプレゼンテーションで、さらなる理解を深めたいと思います！
 - ・視聴者の皆様は、積極的にご質問いただければと思います！

**参加者の皆様お疲れ様でした！入賞者の皆様おめでとうございます！
今後も生成 AI への理解を深め、みんなで社会実装していきましょう！**

※本資料の無断での転載・再配布はご遠慮ください。



SIGNATE
Empowering Your Potential