

# 信頼性/正確性を見える化した RAG 評価フレームワークの提案と検証

伊左次 翔太

NEC ソリューションイノベータ株式会社

sisaji-nes@nec.com

植松 凌太

株式会社 日立製作所

ryota.uematsu.pk@hitachi.com

川西 昂弥

三菱電機ソフトウェア株式会社

kawanishi.koya.ek@mesw.co.jp

濱 憲仁

富士通株式会社

hama.norihito@fujitsu.com

石川 冬樹

国立情報学研究所

f-ishikawa@nii.ac.jp

## 要旨

本稿では、検索拡張生成 (RAG) を用いたシステムについて、信頼性と正確性を見える化した評価フレームワークの提案とその検証について述べる。近年、大規模言語モデル (LLM) の回答精度を向上するための手法として、RAG の需要が増加しており、その品質評価の重要性が一層高まっている。本稿では、ステークホルダーとの継続的な議論・合意形成を想定し、RAG に関する性能や回答不備の特徴・傾向を説明するための評価フレームワークを提案する。また検証実験をとおして、開発した技術要素の妥当性と、提案フレームワークの有効性をそれぞれ確認した。

## 1. はじめに

近年、ChatGPT や Stable Diffusion などの生成 AI 技術が急速に普及し、いわゆる「第 4 次 AI ブーム」に突入したとされている [1]。これに伴い、生成系の大規模言語モデル (Large Language Models: LLM) を活用したさまざまなアプリケーションが注目を集めている。しかしながら、LLM は、学習していない専門性の高いデータ (社内ドキュメントなど) に関する回答精度が著しく低下するという問題を抱える。こうした問題への対策として代表的に挙げられるのが検索拡張生成 (Retrieval-Augmented Generation: RAG) であり、現在では製造業や金融機関、自治体など、分野を問わず RAG を用いたシステムの導入や開発が進められている [2]。専門性の高いデータを扱う上で LLM がハルシネーションを起こ

すリスクも踏まえると、金融機関のように金額の回答を誤ると重大な問題につながるケースが考えられるため、RAG システムの構築では評価の重要性が一層高まる。

RAG の評価については、いくつかのベンチマークやフレームワークが提案されている。一方で、既存の評価手法では、定量評価の精度に主眼が置かれているため、専門知識を持たないステークホルダーに対する説明性に課題が残っている。企業が RAG システムを構築する際には、プロジェクトの目標・方針についての合意形成を図るため、ステークホルダーが納得できるような評価結果や改善施策を説明する必要がある。

本稿では、このような課題を踏まえ、説明性を 3 つの要件に分解し、RAG システムの信頼性と正確性を見える化した評価フレームワークを提案する。提案フレームワークでは、ステークホルダーとの継続的な議論・合意形成を想定し、RAG に関する性能や回答不備の特徴・傾向を説明するための仕組みを提供する。また本稿では、提案フレームワークを実現するために開発した、信頼性スコア・原因分析・質問分析、という 3 つの技術要素を説明する。さらに検証実験のためのシステムを構築し、開発した技術要素が正解として期待する分類を実現していることや、提案フレームワークが定義した説明性の要件を満たしていることを確認した結果について述べる。

本稿の構成は以下のとおりである。2 章では、本稿に関連する背景・課題について述べる。3 章と 4 章では、提案フレームワークとその実現のために開発した技術要素を説明する。5 章で提案フレームワークに対する検証実験の結果を示し、6 章で本稿のまとめと今後の方針について述べる。

## 2. 背景

現在、LLMの回答精度を向上するための手法として、RAGが注目を集めている[3]。RAGは、外部データベースから検索した情報（以降では、コンテキストとよぶ）を用いて、質問に対する回答を生成するため、ハルシネーションを抑制しやすいという特徴を持つ。企業においてもRAGに関連する取り組みが加速しており、チャットボットやナレッジマネジメントなどへの応用を中心として、数多くのアプリケーションが開発されている。

このような背景から、RAGの品質評価の重要性が一層高まっている。研究分野では、包括的な調査論文において、RAGの評価手法がいくつか取り上げられている[4]。RGB (Retrieval-Augmented Generation Benchmark)の研究では、Noise RobustnessやNegative Rejectionなど、合計4つの観点での評価を行なうために、新たなベンチマークを構築している[5]。またRAG評価フレームワークとして、RAGAS (Retrieval Augmented Generation Assessment)やARES (Automated RAG Evaluation System)が提案されている[6, 7]。このうちRAGASは、多角的かつ豊富な指標を備え、現在の性能を定量的に評価できるだけでなく、潜在的な問題点の明確化にも役立つ。

一方、RAGASでは、多数の指標が数値のみで出力されるため、専門知識を持たないステークホルダーに対する説明性に課題がある。具体的には、RAGASの指標として、事実に正確性 (Factual correctness) や意味的類似性 (Semantic similarity) などが算出されるものの、それらの数値について「どの程度良いのか」や「どこを改善すればよいのか」が直感的に伝わりにくい。そのためステークホルダーにRAGの評価結果を理解してもらうためには、RAGASの指標を単純に可視化するだけでは不十分であり、新たな指標や分析技術が必要であると考えられる。

本稿では、上述の課題に取り組むべく、説明性を3つの要件に分解することで、RAGASによる定量評価を活かしつつ、フレームワークとしての補完を図る。これにより、RAGシステムを構築する際、達成すべき目標・改善のための方針が定まりづらいことや、前提条件・評価結果に対する認識相違が生じることなど、ステークホルダーとの議論・合意形成における障壁を下げることをめざす。

## 3. 提案フレームワーク

本章では、前章までに示した課題に取り組むべく、RAGシステムにおける信頼性と正確性を見える化したフレームワークを提案する。

### 3.1 提案フレームワークの全体像

RAGの全体性能を共有してから、詳細項目の議論へとつなげることを想定し、ステークホルダーに対する説明性を以下3つの要件に分解する。

- (1) 性能が一目でわかる
- (2) できないこと・できない原因が明確にわかる
- (3) どこをめざすべきかを議論できる

図1に提案フレームワークの全体像を示す。提案フレームワークでは、各要件に対応する形で、(a) 正確性と信頼性の2つの観点で可視化、(b) 低スコア回答をカテゴリ別に可視化、(c) 可視化と分析根拠を活用した合意形成、という3つの評価フェーズをとおして、RAGシステムを評価・改善する。詳細は後述するが、評価フェーズ(a)では2つの指標により全体性能を示し、評価フェーズ(b)(c)では多様なRAGAS指標を用いることで詳細項目の議論につなげる。

### 3.2 提案フレームワークの詳細

図2に提案フレームワークを実現するシステムの概念構成図を示す。ここでは、以下の機能を連携させることで、RAGの回答生成から評価・分析・可視化までを一連のフローで実行することを想定している。

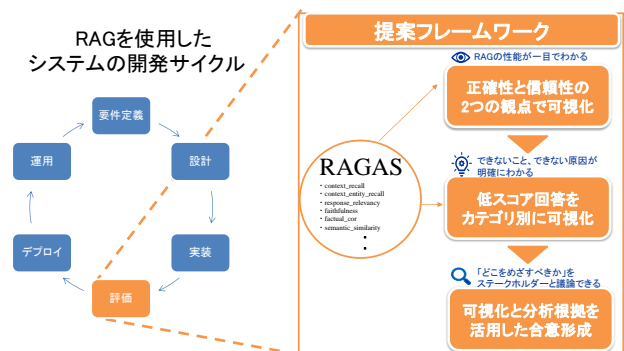


図1. 提案フレームワークの全体像

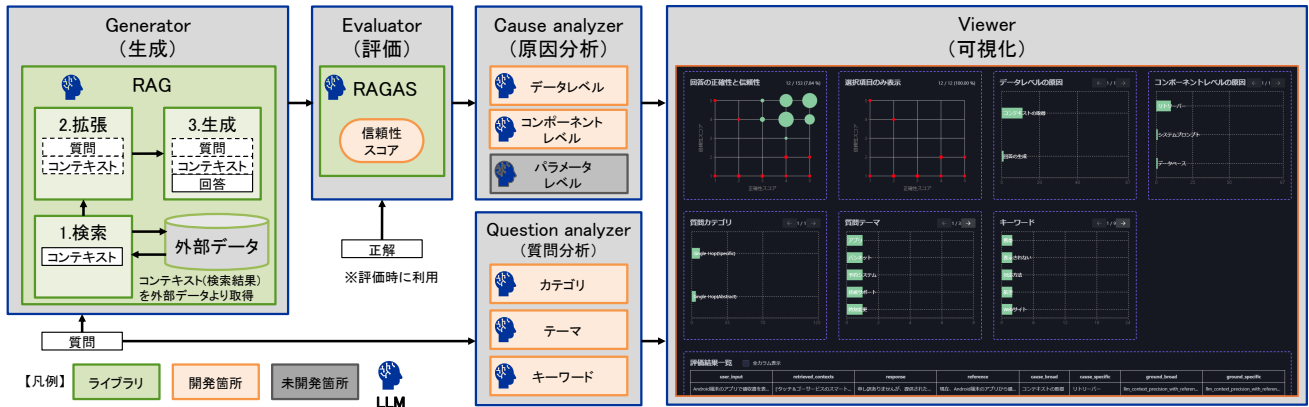


図 2. 提案フレームワークを実現するシステムの概念構成図

- Generator (生成)  
RAG を用いて回答を生成する。また回答を生成する上で、質問に関連するコンテキストを外部データベースから取得する。
- Evaluator (評価)  
RAGAS を用いて回答を評価する。ここでは信頼性を評価する独自作成指標を追加している。
- Cause Analyzer (原因分析)  
データレベル、コンポーネントレベル、パラメータレベルなどの観点で回答不備の原因を分析する。
- Question Analyzer (質問分析)  
カテゴリ、テーマ、キーワードなどの観点で質問の内容を分析する。
- Viewer (可視化)  
評価結果・分析結果をバブルチャートやグラフなどで可視化する。

RAGAS による定量評価のみでは、各指標について「どの程度良いのか」や「どこを改善すればよいのか」が直感的に伝わりにくい。そのためステークホルダーに RAG の評価結果を理解してもらうためには、RAGAS の指標を単純に可視化するだけでは不十分であり、新たな指標や分析技術が必要であると考えられる。提案フレームワークでは、RAGAS の指標を数値的根拠として参考にしつつ、原因分析・質問分析などの結果を可視化することで、説明性の各要件を実現する。以降では、可視化結果に焦点を当てつつ、提案フレームワークの各フェーズについて詳細を説明する。

### 3.2.1 正確性と信頼性の 2 つの観点で可視化

説明性の要件 (1) を満たすため、提案フレームワークでは RAG の性能を正確性・信頼性という 2 つの観点で可視化する。ここでは、RAG の全体性能を一目で把握可能とすることを念頭に、2次元空間での可視化を想定している。RAG はコンテキストをもとに回答を生成する仕組みであり、その性能を可視化するには、(1) 回答と正解の一致度、(2) 回答とコンテキストの関連度、という 2 つの指標を用いるのが好適と考えた。本研究では、前者を正確性スコア、後者を信頼性スコアとよぶ。正確性スコアには、予備実験の結果から、RAGAS 指標であり 5 段階評価の「Rubrics score」を用いている。信頼性スコアとしては、対応する 5 段階評価の指標が RAGAS に存在しないため、後述する独自作成指標を採用する。

図 3 に正確性・信頼性の可視化結果の一例を示す。可視化手法としては、バブルチャートを用いている。ここでは、正確性を横軸 (左側が低スコア、右側が高スコア)、信頼性を縦軸 (下側が低スコア、上側が高スコア) にそれぞれ配置し、バブルの大きさにデータ数を反映させる。赤色のバブルは低スコア回答、緑色のバブルは低スコア

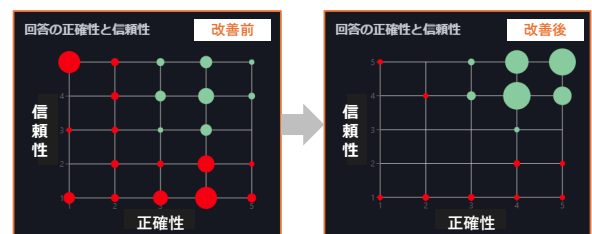


図 3. 正確性・信頼性の可視化結果

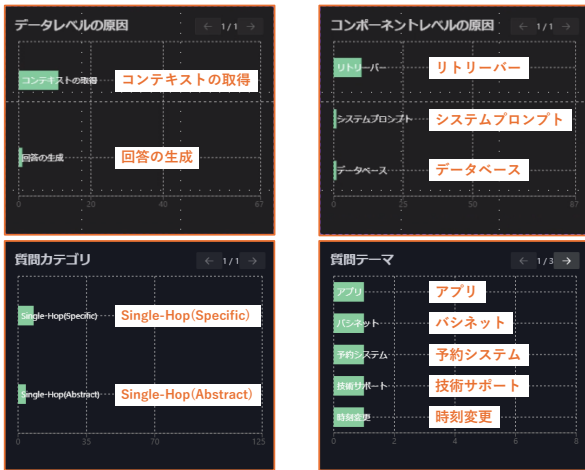


図 4. 原因分析・質問分析の可視化結果

回答以外のデータを意味する。図 3 では、いずれかのスコアが 2 以下のデータを低スコア回答と位置づけており、左枠の改善前から右枠の改善後へと低スコア回答が減少していることを確認できる。この仕組みにより、RAG の全体的な性能を一目で把握可能とする。

### 3.2.2 低スコア回答をカテゴリ別に可視化

説明性の要件 (2) を実現するため、提案フレームワークでは低スコア回答をカテゴリ別に可視化する。ここで低スコア回答を分類するために、LLM を用いた原因分析・質問分析を行う。原因分析では、RAGAS の評価結果 (Context Precision, Context Recall, Faithfulness など) を用いて、低スコア回答となった「原因」を提示する。また原因分析については、タスク分割により LLM の精度が向上しやすいという考えのもと、データレベル・コンポーネントレベル・パラメータレベル、という 3 つの観点に分けて実施する。質問分析では、回答不備の特徴・傾向を明らかにするため、質問の「カテゴリ」「テーマ」「キーワード」を抽出する。なお原因分析・質問分析の詳細は、次章にて説明する。

図 4 に原因分析・質問分析の可視化結果の一例を示す。可視化手法としては、棒グラフを用いている。ここでは、分類件数の多い順に原因や質問カテゴリなどを表示する。またバブルチャートの選択箇所に絞り込んで分類件数を示す機能も備えており、着目したデータ群の特徴を確認することも可能である。図 4 の例では、選択した低スコア回答において、データレベルの原因として

「コンテキストの取得」、コンポーネントレベルの原因として「リトリーバー」が最多の分類件数となっていることがわかる。また質問カテゴリとしては、「Single-Hop (Specific)」と「Single-Hop (Abstract)」に分かれていることが確認できる。このような可視化結果により、回答不備の特徴・傾向を分析可能とする。

### 3.2.3 可視化と分析根拠を活用した合意形成

説明性の要件 (3) に関しては、可視化と分析結果を活用することで、ステークホルダーとの合意形成を図ることを想定している。上述の可視化結果により、現在の性能や回答不備の特徴・傾向をステークホルダーと共有し、各マイルストーンにおいて「何を目標に」「どのような方針で」施策を講じるのか、といった議論につなげる。また原因分析では、低スコア回答となった「原因」だけでなく、RAGAS の評価結果を踏まえた「分析根拠」を提示することも可能である。提案フレームワークでは、この分析根拠を提示することで、詳細項目を議論して改善方針を決定できるようにしている。例えば「悪天候や自然災害の影響で搭乗予定便が欠航になった場合の払い戻し手続きが可能であることが明確に記載されていますが、生成された回答は払い戻しができないと述べています」といった分析根拠を提示することで、金銭に関する回答は工数を積んで対応するという方針を決めることが考えられる。以上のフレームワークにより、低スコア回答の影響範囲と対策の優先度を考慮した議論を実現する。

## 4. 技術要素

### 4.1 信頼性スコア

この信頼性スコアは、LLM を用いることで、最も低い「1」から最も高い「5」までの 5 段階で評価される。信頼性スコアは、Rubrics score の実装を基礎として、定義や記述などを一部改変することで作成している。特に 5 段階評価の基準となるスクリプトを回答とコンテキストの関連度が分類できるように新しく定義する。具体的には「The response is fully based on the contexts, (後略)」のようなスクリプトを評価値ごとに与えている (ここで「fully」の箇所は、対応する評価値に応じて、「not」「partially」「moderatory」「mostly」に置き換わる点に注意)。このような基準を設定することで、信頼性スコアの算出を LLM で実現している。

## 4.2 原因分析

原因分析は、RAGAS による定量評価に基づき、回答不備の原因を分類する技術である。RAGAS の指標のみでなく、質問・コンテキスト・回答・正解の情報を入力とすることで、「回答不備がなぜ生じたのか」や「不足している情報は何か」といった分析根拠まで提示できるように工夫している。本稿では、データレベル・コンポーネントレベルという、分類の観点異なる 2 種類の原因分析を扱う。データレベルの観点では、質問の内容、コンテキストの取得、回答の生成、正解の定義、不備なしが分類項目となる。コンポーネントレベルの観点では、検索クエリ、データベース、リトリーバー、生成モデル、システムプロンプト、後処理、正解の定義、不備なしが分類項目にあげられる。パラメータレベルの観点について、チャンクサイズやオーバーラップなどを想定しているが、ステークホルダーに開示する情報としては粒度が細かいため、本研究では開発の対象外とした。

原因分析を LLM で実現するにあたり、プロンプトには以下の情報を含めている。

- (1) 役割の設定: 回答不備の原因特定をタスクに設定
- (2) データの埋込: 質問、コンテキスト、回答、正解、および RAGAS の評価結果 (指標の数値) を埋込
- (3) データカラムの説明: データのカラムを説明
- (4) 分析手順の記述: 分類項目ごとに判定条件・判定基準を定め、分析する手順を記述
- (5) 再検証の指示: 原因が分析の結果から忠実に導かれているかを再検証し、適宜修正するように指示
- (6) 出力形式の指定: JSON 形式での出力を指定

ここでは、予備実験の結果をもとに、出力結果の安定していた「CoT (Chain of Thought)」をプロンプトに採用している。具体的には、分析手順の記述において、RAG の動作順序に着目し、入力側から優先して原因を特定するように指示を与える。このとき、分類項目を明示的に与えることで、出力結果の表記ゆれを防ぐような設計としている。また判定条件 (判定対象を選別する条件) において RAGAS の指標を用いるものの、最終的な判定基準としては質問・コンテキスト・回答・正解を参照した総合的な判断とすることで、RAGAS の精度への依存度を下げている。以上のような仕組みにより、回答不備の原因を特定することで、RAG の改善点を明らかにすることにつながる。

## 4.3 質問分析

質問分析は、RAG の入力となる質問の内容を分類する技術である。本稿では、カテゴリ・テーマ・キーワードという、分類の観点異なる 3 種類の質問分析を扱う。このうちカテゴリの観点では、Single-Hop (Specific), Single-Hop (Abstract), Multi-Hop (Specific), Multi-Hop (Abstract) が分類項目となる。またテーマ・キーワードの観点については、分類項目がデータセットごとに異なる。

質問分析を LLM で実現するにあたり、プロンプトには以下の情報を含めている。

- (1) 役割の設定: 質問内容の分類をタスクに設定
- (2) データの埋込: 質問の情報を埋込
- (3) 分類ルールの記述: 分類項目ごとに定義や例示を与え、分類のためのルールを記述
- (4) 注意事項の指示: 質問内容の分類を実施する上で、特に注意が必要な項目を指示
- (5) 出力形式の指定: JSON 形式での出力を指定

ここでは、分析に有効な分類項目がタスクや要求により異なる想定のもと、定義や例示を与えるだけのシンプルなプロンプトを採用している。例えば、カテゴリの観点での質問分析では、RAGAS の公式サイト [8] における Query type を参考に、質問の難易度を確認できる分類項目を採用したが、別の観点 (例えば、質問不備によるリスク度など) で分析を行いたい場合は、(3) 分類ルールの記述における定義や例示を変更することで対応可能な構成としている。以上のような仕組みにより、質問を分類することで、回答不備の特徴・傾向を明らかにすることにつながる。

## 5. 検証実験

提案内容の妥当性や有効性を評価するために「技術要素の評価」と「提案フレームワークの評価」の 2 種類の検証を行った。ここでは、各検証の目的・条件・結果・考察について詳述する。

### 5.1 技術要素の評価

技術要素の評価では、「信頼性スコア」と「原因分析」の 2 項目を検証する。なお本評価では、Hugging Face で

表 1. 信頼性スコアの評価結果（太字は各行で正解として期待するスコア群）

コンテキストの種類	Score 5	Score 4	Score 3	Score 2	Score 1
理想的なコンテキスト	<b>30</b>	<b>20</b>	0	0	0
冗長なコンテキスト	<b>30</b>	<b>20</b>	0	0	0
欠損したコンテキスト	2	4	10	<b>9</b>	<b>25</b>
関連なしコンテキスト	0	0	0	<b>0</b>	<b>50</b>

公開されている WikiEval データセット [9]（50 件の質問・コンテキスト・回答・正解のセット）を用いる。

### 5.1.1 信頼性スコアの妥当性評価

1 つ目の技術評価の目的は、独自作成した指標である信頼性スコアの妥当性を確認することである。評価の条件として、以下の 4 種類のケースを想定したデータを用意し、信頼性スコアが各ケースで期待する評価値となるかどうかを検証する。

- (1) 理想的なコンテキストから正解が生成
- (2) 冗長なコンテキストから正解が生成
- (3) 欠損したコンテキストから正解が生成
- (4) 関連なしコンテキストから正解が生成

これらのうち、ケース (1) (2) は信頼性スコアが高評価 (Score 5 または Score 4) となること、ケース (3) (4) は信頼性スコアが低評価 (Score 2 または Score 1) となることが期待される。なおケース (3) (4) は、WikiEval データセットを一部改変することで作成した。

信頼性スコアの評価結果を表 1 に示す。ここで評価は RAGAS に信頼性スコアを追加することで実施した。表 1 から、理想的・冗長なコンテキストを用いたケース (1) (2) では、正解に必要な情報を含むため、信頼性スコアが高評価 (Score 5 または Score 4) を示している。また欠損した・関連なしコンテキストを用いたケース (3) (4) では、正解に必要な情報を含まず、低評価 (Score 2 または Score 1) となったものが多い。なお評価結果の詳細を確認したところ、ケース (3) で高評価 (Score 5 または Score 4) となった 6 件のデータの評価については、回答に必要な情報が欠損したコンテキストにも含まれており、妥当な結果であることを確認した。以上の評価から、信頼性スコアが、回答とコンテキストの関連度について、一定の傾向を捉えているものと考えられる。

### 5.1.2 原因分析の妥当性評価

2 つ目の技術評価の目的は、データレベル・コンポーネントレベルという 2 種類の原因分析の妥当性を確認することである。評価の条件として、以下の 6 種類のケースを想定した RAG を用意し、各ケースで期待する項目を原因として特定できるかどうかを検証する。

- (1) 理想形（正解と回答が一致）
- (2) 標準形（意図的な不備なし）
- (3) データベースに不備あり
- (4) リトリーバーに不備あり
- (5) 生成モデルに不備あり
- (6) システムプロンプトに不備あり

これらのうち、ケース (3) (4) (5) (6) が主要な評価対象であり、それぞれの RAG に埋め込んだ不備を原因として特定することが期待される。

データレベル・コンポーネントレベルの観点での原因分析の評価結果を表 2 と表 3 にそれぞれ示す。まず、表 2 が示すデータレベルの評価結果では期待通り、データベース・リトリーバーに不備がある場合に「コンテキストの取得」が多く特定され、生成モデル・システムプロンプトに不備がある場合に「回答の生成」が多く特定された。また、表 3 が示すコンポーネントレベルの評価結果でも、各ケースで RAG の特徴に合った原因（データベースやリトリーバーなど）が最も多く特定された。以上の評価から、原因分析全体として一定の妥当性があるものになっていると考えられる。一方、ケース (3) でリトリーバーへの誤分類 8 件、ケース (6) で生成モデルへの誤分類 10 件がそれぞれ確認された。RAGAS の粒度がデータレベルに相当するため、追加の入力情報を用いる、といった対策が必要であると考えられる。また以上の評価では、RAG に明確な不備があるケースを扱った。実応用を想定すると、明確な不備がある状況は稀であるため、より本番に近い環境での評価も今後の課題となる。

表 2. 原因分析（データレベル）の評価結果（太字は各行で正解として期待する原因）

RAG の特徴	不備なし	質問の内容	コンテキストの取得	回答の生成	正解の定義
理想形（正解と回答が一致）	45	0	1	4	0
標準形（意図的な不備なし）	30	0	2	13	5
データベースに不備あり	2	0	<b>48</b>	0	0
リトリーバーに不備あり	1	0	<b>48</b>	0	1
生成モデルに不備あり	10	0	1	<b>38</b>	1
システムプロンプトに不備あり	1	0	0	<b>49</b>	0

表 3. 原因分析（コンポーネントレベル）の評価結果（太字は各行で正解として期待する原因）

RAG の特徴	不備なし	DB	RT	GM	SP	その他
理想形（正解と回答が一致）	49	1	0	0	0	0
標準形（意図的な不備なし）	41	3	0	0	3	3
データベース（DB）に不備あり	2	<b>40</b>	8	0	0	0
リトリーバー（RT）に不備あり	1	0	<b>48</b>	0	0	1
生成モデル（GM）に不備あり	13	2	0	<b>35</b>	0	0
システムプロンプト（SP）に不備あり	3	0	0	10	<b>37</b>	0

## 5.2 提案フレームワークの評価

提案フレームワークの評価としては、定義した説明性の要件を満たしているかを検証することを目的とする。具体的には、回答不備が賠償問題などにつながるリスクが高いドメインとして航空会社を想定し、航空会社の QA チャットボットを開発するシナリオでデータセットを作成した。データセットの概要は、以下のとおりである。

- ・ 質問：JAL 公式ページ [10] の「よくある Q&A」の国内線に関する質問（153 件）
- ・ 正解：上記質問に対する回答（153 件）
- ・ データベース：上記回答をまとめたテキストファイル

この条件で提案フレームワークを適用し、(1) 性能が一目でわかる、(2) できないこと・できない原因が明確にわかる、(3) どこをめざすべきかを議論できる、という 3 点を検証した。その結果は以下のとおりである。

### 5.2.1 性能が一目でわかる

本検証では、RAG のデータベースのデータ量を変更した場合と、生成モデルを変更した場合のバブルチャ

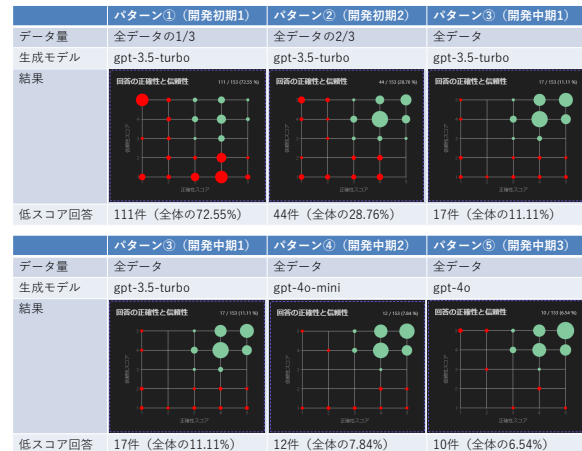


図 5. 説明性の要件 (1) に対する検証内容

トの変化を確認することで、性能の差異が一目でわかるかを検証した。図 5 にデータ量を変更した場合（上段 3 つ）、生成モデルを変更した場合（下段 3 つ）のバブルチャートを示す。本検証では、正確性スコアと信頼性スコアのいずれかが 2 以下の回答を低スコア回答であるとステークホルダーと合意したと仮定し、赤色のバブルとしている。低スコア回答数がモデルの性能向上に伴い、

減少していることから、RAG の性能確認に正確性スコアと信頼性スコアは有効と考える。

また、低スコア回答の傾向について、(i) 正確性スコアと信頼性スコアのどちらも低い場合、(ii) 正確性スコアのみ低い場合、(iii) 信頼性スコアのみ低い場合に分けて確認した。低スコア回答数が最も多いパターン (1) を対象に、回答に明確な誤りがあるかと、回答にコンテキストに情報が存在しないことの記載があるかの2つの観点で確認した結果を表 4 に示す。(i) には可能を不可と回答するような明確な誤りが多く、(ii) にはコンテキストに情報が存在しないことを記載している回答(「申し訳ありませんが、提供されたコンテキストには(中略)に関する情報は含まれていません。(後略)」)のような回答が多くみられた。(iii) には明確な誤りが少ないものの、LLM が持つ知識で作成したと考えられる具体性にかける回答が多くみられた。以上のことから、正確性スコアと信頼性スコアは低スコア回答の傾向確認にも有効と考える。本検証では、Generator にコンテキストを参照する最低限の指示のみを行い、自社に関する質問に回答させた。今後は、Generator により多くの指示(例えば、コンテキストに情報が存在しない場合の指示)を行った場合や、他社に関する質問やセンシティブな質問といった多様なデータでも評価していく必要がある。

### 5.2.2 できないこと・できない原因が明確にわかる

次にデータレベルの原因から「できないこと」、コンポーネントレベルの原因から「できない原因」が確認可能かを検証する。図 6 に、5.2.1 項のパターン (4) の低スコア回答を対象とした、原因分析・質問分析の可視化結果を示す。低スコア回答 12 件のうち、リトリーバーが原因のものが 10 件あるため、リトリーバーを改良することで、RAG の性能改善を見込めると予測できる。また、質問カテゴリを確認すると、「Single-Hop (Specific)」カ

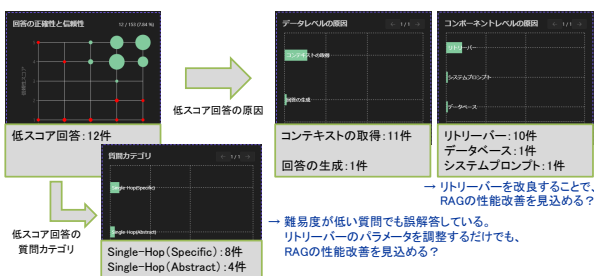


図 6. 説明性の要件 (2) に対する検証内容

テゴリに属する質問が 8 件あり、難易度が低い質問でも誤回答していることがわかる。このことから、リトリーバーのパラメータを調整するだけでも RAG の性能改善を見込めると予測できる。

表 5 にリトリーバーのパラメータ変更前後における低スコア回答数の内訳を示す。ここでリトリーバーのパラメータは以下のように変更した。

- ・ 組み込みモデル: text-embedding-ada-002 から text-embedding-3-large に変更
- ・ Chunk size: 1000 から 800 に変更
- ・ Chunk overlap: 0 から 200 に変更
- ・ Search kwargs:  $k = 3$  から  $k = 4$  に変更

ここで Search kwargs の  $k$  は取得するコンテキストの数を意味している。表 5 が示すように、リトリーバーのパラメータを変更することで、実際に RAG の性能が改善することを確認した。このことから、データレベルの原因から推測した「できないこと」、コンポーネントレベルの原因から推測した「できない原因」は正しいといえる。なお、リトリーバーのパラメータを変更したことでスコアが低下した回答もあった。このことから、改良前後の差分に気づける仕組みも必要になると考える。

### 5.2.3 どこをめざすべきかを議論できる

本検証では、可視化と分析根拠を活用することで、ステークホルダーと「どこをめざすべきか」を議論できるかを検証する。図 7 に、5.2.2 項のリトリーバー改良後の結果のうち、正確性スコアと信頼性スコアのどちらも低い 2 つのケースの例を示す。

例 1 (正確性スコアが 2, 信頼性スコアが 1 のケース) は、「バシネットを利用することはできますか?」という質問に対し、コンテキストに情報が存在せず回答できていないケースである。データレベルの原因および分析根拠を確認すると、バシネットに関する情報を取得できていないことがわかり、正しく回答できるようにするためには、リトリーバーのさらなる改良が必要になると予測できる。例 2 (正確性スコアが 2, 信頼性スコアが 2 のケース) は、「悪天候や自然災害の影響で搭乗予定便が欠航になった場合、(中略) 航空券の払い戻し手続きはできますか?」という質問に対し、払い戻し可能にもかかわらず払い戻し不可と回答したケースである。コンポーネ

表 4. パターン (1) の低スコア回答の傾向 (太字は各行でスコアの定義から期待する項目)

正確性スコア	信頼性スコア	明確な誤りがある件数	情報が存在しない記載がある件数	該当なしの件数
低 (1~2)	低 (1~2)	<b>10</b>	3	0
低 (1~2)	高 (4~5)	1	<b>32</b>	0
高 (4~5)	低 (1~2)	7	0	<b>41</b>

表 5. リトリバーのパラメータ変更前後の低スコア回答数 (太字はパラメータ変更の効果を期待する項目)

データレベルの原因	コンポーネントレベルの原因	質問カテゴリ	変更前の件数	変更後の件数
コンテキストの取得	リトリバー	Single-Hop (Specific)	<b>7</b>	<b>1</b>
		Single-Hop (Abstract)	<b>3</b>	<b>2</b>
コンテキストの取得	データベース	Single-Hop (Specific)	0	1
		Single-Hop (Abstract)	1	0
回答の生成	システムプロンプト	Single-Hop (Specific)	1	1
		Single-Hop (Abstract)	0	0

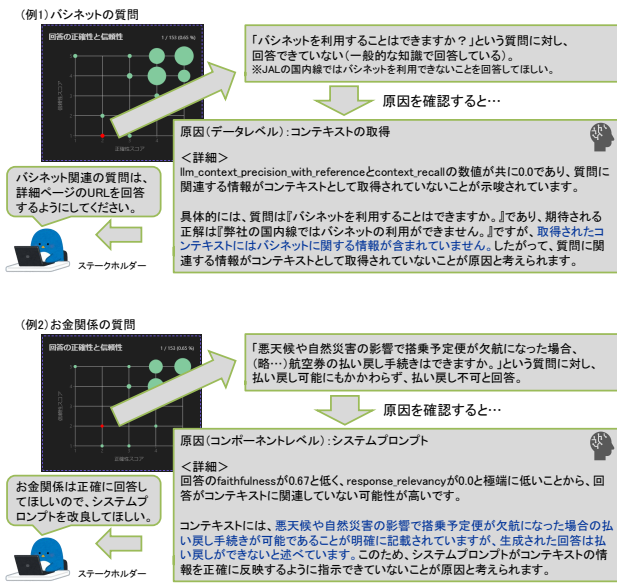


図 7. 説明性の要件 (3) に対する検証内容

ントレベルの原因および分析根拠を確認すると、コンテキストに基づいて回答を生成できていないためシステムプロンプトを見直すべきことがわかる。

このような回答不備の原因や処置方針を説明したうえで、ステークホルダーの持つ情報(重要度や予算など)と合わせて考慮することで、例えば、例1は工数が少

なく済む対応(バシネット関連の質問には詳細ページのURLを回答するなど)にとどめ、例2は金銭に関する質問であるため、工数を積んで対応するといった議論が実現すると考える。以上のことから、可視化と分析根拠の活用はステークホルダーとの合意形成に有効と考える。なお、質問分析の質問カテゴリを確認したところ、本評価に用いたデータセットの質問のほとんどはSingle-Hop(単一情報から回答可能)に分類されていた。実応用を想定すると、Multi-Hop(複数の情報を統合しなければ回答できない)に属する質問でも評価していく必要があると考える。

## 6. おわりに

本稿では、専門知識を持たないステークホルダーに対する説明性を向上するために、信頼性と正確性を見える化したRAG評価フレームワークを提案した。提案フレームワークでは、ステークホルダーとの継続的な議論・合意形成を想定し、RAGに関する性能や回答不備の特徴・傾向を説明するための仕組みを提供する。検証実験のためのシステムを構築した結果、開発した技術要素が正解として期待する分類を実現していることや、提案フレームワークが定義した説明性の要件を満たしていることを確認した。提案フレームワークは、RAGASによる

定量評価を活かしつつ、説明性を高める技術要素を補完した、より実践的な枠組みであると考えられる。

今後の展望としては、検証実験でも述べたように、技術要素の高度化・検証や提案フレームワークの拡張の観点から、以下のような取り組みが考えられる。

- ・原因分析の精度向上・追加実装  
コンポーネントレベルの原因分析では、データベース・リトリーバー、生成モデル・システムプロンプトに関する原因をより正確に切り分け、分類することが課題となる。またパラメータレベルの原因分析を追加実装し、チャンクサイズやオーバーラップなど、より細かい粒度で改善点を明らかにすることも重要となる。
- ・多様なデータでの評価  
複数の情報を統合しなければ回答できない質問や不備が明確でない RAG に対しての検証を行うことが必要と考える。
- ・RAG 改良前後の差分の比較  
改良前は回答できていた質問が、改良後は回答できなくなるケースに気づける仕組みを導入することが一案となる。

上記の取り組みをとおして、より説明性の高い評価フレームワークを実現し、ステークホルダーが信頼・納得できる RAG システムの構築をめざしていく。

## 謝辞

本研究は、国立情報学研究所の社会人向け教育プログラムであるトップエスエーのソフトウェア開発実践演習において取り組まれたものである。ここに記して謝意を表する。

## 参考文献

- [1] 総務省, “令和 6 年版情報通信白書 (PDF 版),” <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r06/pdf/n1320000.pdf> (参照日 2025 年 2 月 2 日)
- [2] 独立行政法人 情報処理推進機構, “テキスト生成 AI の 導入・運用ガイドライン,” <https://www.ipa.go.jp/jinzai/ics/core->

[human\\_resource/final\\_project/2024/f55m8k0000003spo-att/f55m8k0000003svn.pdf](https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r06/pdf/n132000003spo-att/f55m8k0000003svn.pdf) (参照日 2025 年 2 月 2 日)

- [3] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, “Retrieval-augmented generation for knowledge-intensive nlp tasks,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459-9474, 2020.
- [4] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, “Retrieval-augmented generation for large language models: A survey,” *arXiv preprint arXiv:2312.10997*, 2, 2023.
- [5] J. Chen, H. Lin, X. Han, and L. Sun, “Benchmarking large language models in retrieval-augmented generation,” *In Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17754-17762, 2024.
- [6] S. Es, J. James, L. Espinosa-Anke, and S. Schockaert, “Ragas: Automated evaluation of retrieval augmented generation,” *arXiv preprint arXiv:2309.15217*, 2023.
- [7] J. Saad-Falcon, O. Khattab, C. Potts, and M. Zaharia, . “Ares: An automated evaluation framework for retrieval-augmented generation systems,” *arXiv preprint arXiv:2311.09476*, 2023.
- [8] “Testset Generation for RAG,” Ragas, [https://docs.ragas.io/en/stable/concepts/test\\_data\\_generation/rag](https://docs.ragas.io/en/stable/concepts/test_data_generation/rag) (参照日 2025 年 3 月 8 日)
- [9] Exploding Gradients, “WikiEval,” Hugging Face, <https://huggingface.co/datasets/explodinggradients/WikiEval> (参照日 2025 年 2 月 6 日)
- [10] JAPAN AIRLINES, “国内線・国際線・JAL マイレージバンクに関する Q&A,” <https://faq.jal.co.jp> (参照日 2025 年 2 月 6 日)