失敗知識データベースを活用した RAG 評価用 Why 型 QA データセットの自動生成

野澤遼太郎 [1] 林瑛勳 [2] 玖津見航 [2] 高田尚輝 [2] 森辰則 [2]
[1] 横浜国立大学 [2] 横浜国立大学大学院
nozawa-ryotaro-dw@ynu.jp lim-younghun-sw@ynu.jp kutsumi-ko-sr@ynu.jp
takada-naoki-xs@ynu.jp tmori@ynu.ac.jp

概要

RAG システムの性能評価には、高品質な QA データセットの存在が必要不可欠である。しかし、従来の手動作成では膨大な労力と時間が必要となるため、自動化による生成手法が注目されている。また、Where 型や When 型、What 型などの factoid 型で、回答が短いテキストになる QA については、データセットを自動生成する手法が提案されている一方、「なぜ」や「どうして」、「どのように」といった Why 型や How 型のような、non-factoid 型で回答が比較的長いテキストになる QA については、データセットの自動生成の研究が十分になされていない。本研究では、Why 型に特化した QA データセットを自動生成する手法を提案する。評価実験の結果、Why 型の質問形式かつ一定程度の品質を持つデータセットが生成できることを確認した。

1 はじめに

RAG（Retrieval Augmented Generation）は、情報検索と大規模言語モデル (LLM) による生成を組み合わせたシステムであり、外部データを活用することで正確かつ柔軟な回答を生成できることから注目されている [1]。RAG は、LLM の生成のみでは対応しきれない事実性の確保やドメイン知識の補完といった問題に対する有効な解決策として認識されている [2]。RAG システムの性能を適切に評価するためには、高品質な QA データセットが必要であるものの、人手による QA データセット作成には次のような課題が存在する [3]。

作成コストの高さ：手動による QA データセットの構築は一般的に時間的・人的リソースの消費が大きく、特にドメイン知識を要する実案件では、専門家の確保や品質を確認のためにさらに多大なコストが必要となる。

ドメイン依存性：汎用的な性能評価のためには、複数のドメインにおける多様なデータセットが必要になるが、人手では作成コストが莫大になるため現実的ではない。

データの更新容易性：実運用では QA の対象や内容が時間の経過とともに変化するため、評価データセットも随時更新を行う必要があるが、人手更新の運用コストは大きな負担となる。

また、RAG システムの評価は依然として困難であり、特に長文回答の厳密な正確さを定義することが困難な non-factoid 型質問において顕著である [4]。

これらを解決するため、本研究では LLM を用いて Why 型の QA データセットの自動生成に取り組む。

2 関連研究

寺井ら [3] は、RAG 評価用データセットに求められる品質観点を整理した上で、高品質なデータセットの自動生成手法を提案している。彼らの手法は、QA ペアデータ生成、品質判定によるフィルタリング、パラフレーズによるペアの拡張という三段階のプロセスから構成される。QA ペア生成時には 5W1H の質問観点を用いたオープンクエスチョン（回答が自由記述形式の質問）やクローズドクエスチョン（回答が選択肢形式の質問）を指定することで多様性に配慮し、質問者の立場に関する説明文を与えることで現実性の向上を図っている。

しかし、寺井らの手法では、生成された質問型の分布に偏りがあり、factoid 型である What 型の割合が 37.5%、When 型が 15.0%であるのに対し、non-factoid 型の Why 型は、5.0%、How 型は 15.0%と Why 型の生成割合が少ないという特徴が見られた。

本研究は、寺井らの品質観点に基づくアプローチ

を踏襲しつつ、Why 型質問に特化した QA データセットの自動生成に取り組む点で新規性を有する。具体的には、失敗知識データベース [5] という人手で構造化された知識ベースを活用することで回答の信頼性を担保しつつ、プロンプトによる質問タイプの厳密な制御によって Why 型質問のみを生成する手法を提案する。これにより、RAG システムの因果関係理解能力を評価するための専門的なデータセットの構築が可能となる。

3 提案手法

3.1 手法の概要

本研究では、失敗知識データベースの構造化された情報を活用し、Why 型質問に特化した QA データセットを自動生成する手法を提案する。提案手法は、前処理による事例概要文章中の原因を意味する文の特定、LLM を用いた質問生成、QA ペアの構築という三段階のプロセスから構成される。得られるデータセットは、1)RAG の検索対象としての、事例概要文書集合、ならびに、2)QA ペアの集合、の組である。ある質問 Q に対する回答 A は、1) の文書集合中のいずれかの文書に現れる表現であり、質問 Q の回答にふさわしいものである。本手法の特徴は、人手で構造化された失敗知識データベースを出発点とすることで、回答の信頼性を担保しつつ、Why 型質問の自動生成を実現する点にある。

3.2 使用データ

本研究では、失敗学会が公開する失敗知識データベース [6] を使用する。失敗知識データベースは、科学技術分野における失敗事例を「事例名称」「事例概要」「事象」「経過」「原因」「対処」「背景」などの属性に分類して蓄積した構造化データセットである。それぞれの項目は、複数の文から構成されるテキストである。「事例概要」は、他の項目の内容を網羅した文章となっているが、文章中のどの部分がどの項目に対応するかといった情報はない。本研究では、失敗知識データベースからスクレイピングを行い、各項目を JSON 形式の構造化データに成形したものを入力データとして使用する。自動生成の評価実験においては、「機械」「化学」「石油」など計 16 分野のうち、機械分野の事例 210 件を対象とした。

3.3 提案手法の手順

Step 1：意味的類似度に基づく原因文の特定（前処理） Step 1 では、処理対象とする事故事例文書の各々について、「事例概要」中のどの文が「原因」のテキストと関連が高いかを特定する前処理を行う。この処理により、Why 型質問の回答となる原因文を事例概要から抽出することが可能となる。

具体的には、Sentence-BERT を用いて事例概要の各文と他の各項目の各文との間の意味的類似度を計算する。使用するモデルは sentence-bert-base-ja-mean-tokens-v2-finetuned である。ある失敗事例における事例概要の各文に対し、最も類似度が高い他項目の文を選び、その項目名を事例概要の各文のラベルとした。さらに、「原因」をラベルとして持つ事例概要文のうち、先の類似度計算の値が最も高かったものを、当該失敗事例の事例概要における原因を説明する文として抽出する。これにより、事例概要文中で最も原因を説明している箇所を特定する。この前処理により、210 件の事例から 188 件が抽出された。

Step 2：LLM を用いた Why 型質問の生成 Step 2 では、Step 1 で特定した、事故原因を表しているであろう事例概要の文を回答として、その回答が正解となるような質問を LLM で生成する。

質問生成には GPT-5 を使用し、以下の入力を与え、適切な質問 Q の生成を要求する：

- **文脈 C**：事例概要の全文
- **回答 A**：Step 1 で抽出した原因に対応する事例概要の文

プロンプトには、適格な Why 型質問を生成するための以下の制約条件を設定した。

制約条件 1：質問タイプの明確な指示（Why 型限定）生成する 5 つの質問はすべて、文法的に「Why 型（なぜ〜したのか？）」とすることを指示する。「〜の原因は何か？」「〜の理由は何か？」といった What 型の形式は明示的に禁止し、「なぜ、〜という事態になったのか？」「どうして、〜が発生したのか？」「どのような理由で、〜に至ったのか？」といった推奨フォーマットを提示する。また、質問の対象（Why の焦点）を事故そのものの発生理由、特定の現象の発生理由、背景要因の 3 種類に変えてバリエーションを持たせることを指示する。

制約条件 2：指示語の禁止（単独の質問としての一貫性の確保）「この」「その」「あの」「例の」などの指示語・代名詞を一切使用しないことを指示する。質問文単体で意味が通じるよう、指示語の代わりに具体的な名詞を使用することを求める。

制約条件 3：メタ表現の禁止「文中」「本文」「原文」「著者は」「下線部は」といった、テキストそのものを指す言葉を一切使用しないことを指示する。

制約条件 4：図表・視覚情報への言及禁止「図 1」「表 2」「グラフ」といった、テキスト以外の視覚情報を指す言葉を一切使用しないことを指示する。

制約条件 5：事例の具体化どの事例について聞いているかが一意に定まるよう、文脈に含まれる固有名詞（正式名称）、日付、場所、数値を質問文に明記することを指示する。

各事例に対して 5 つの質問を生成し、合計 940 件の質問 Q を生成した。

Step 3：QA ペアの構築　Step 3 では、Step 2 で生成された質問 Q の各項目と、質問 Q 生成時に使用した回答 A を組として、QA ペアの集合を構築する。その QA ペア集合と事例概要文書集合の 2 つをもって、最終的な QA データセットを構築する。

4　評価実験

提案手法を用いた QA データセットの生成および評価を実施し、生成された質問の現実性と質問タイプの観点から提案手法の有効性を検証した。

4.1　実験概要

QA データ生成の対象として、失敗学会の失敗知識データベースから機械分野の事例 210 件を使用した。前処理の段階で事例概要中に原因を意味する文が存在しない事例を除外し、最終的に 188 件の事例を対象とした。質問生成には GPT-5 を使用し、各事例に対して 5 つの Why 型質問を生成することで、合計 940 件の QA ペアを生成した。

有効性検証では、生成した QA データセットに対して 2 つの評価を実施した。1 つは QA ペアとしての品質検証、もう 1 つは質問タイプの検証である。

4.2　評価方法

4.2.1　生成された質問の品質検証

生成した QA データセットの品質検証では、Xiao ら [7] が提唱する RAG 評価用データセットに求められる品質観点に基づき評価を行った。以下は Xiao らによる品質観点の定義を寺井ら [3] が日本語訳したものである。

現実性（Realism）：データセットは現実のユースケースを反映したものである必要がある。

信頼性（Reliability）：ベンチマークの正答のデータは正確でなければならず、質問と回答の妥当性が担保されている必要がある。

多様性（Richness）：ベンチマークとなるデータセットは、様々な質問タイプやユースケースをカバーしている必要がある。

洞察性（Insightfulness）：ベンチマークとなるデータセットは、種類や難易度への対応などソリューションの持つ性能を多角的に評価できるものである必要がある。

持続性（Longevity）：ベンチマークのシナリオとデータはすぐに陳腐化することなく、定期的に更新されている必要がある。

本研究ではこれらの観点のうち、現実性について著者による主観評価を実施した。生成した QA ペアの中からランダムに 40 件を抽出し、以下の基準で二値評価を行った。

現実性の評価基準: 事例概要の内容に基づき、実際に発生し得る質問であるかを各 QA データについて判定し、その適合割合を現実性の評価とする。

なお、信頼性については、本手法では前処理により構造化された失敗知識データベースから原因を意味する事例概要文を抽出し、これを回答 A としているため、一次情報に基づく正確な回答が 100%担保される。そのため、信頼性の評価は実施しなかった。また、多様性については、本研究では Why 型の質問生成に特化しているため、評価対象外とした。洞察性についても、システムの多角的な性能評価を実現するために様々な種類や難易度の QA ペアを生成する必要があるが、難易度の明確な定義や生成手法の確立が今後の課題であるため、本研究では評価対象外とした。持続性については、入力文書自体の更新頻度や陳腐化に関する指標であり、QA ペアの生成手法そのものとは直接関係しないため、評価の対象外とした。

4.2.2　質問タイプ制御の検証

生成された質問が確実に Why 型であるかを検証するため、LLM を用いた質問タイプの自動分類を実施した。940 件すべての質問に対して、GPT-5 を使用し 5W1H（Who, When, Where, What, Why, How）

表 1　データセット品質検証結果（現実性）

評価観点	適合率 (%)
現実性	67.5

表 2　質問タイプ分布（5W1H）

質問タイプ	該当件数/全件数	割合 (%)
Why	940/940	100.0
Who	0/940	0.0
When	0/940	0.0
Where	0/940	0.0
What	0/940	0.0
How	0/940	0.0

のいずれに該当するかを判定した。判定の際には、文中に「場所」や「人物」を示す単語が含まれていても、質問の意図が「理由（なぜそうなったか）」であれば必ず Why に分類するよう指示した。

4.3　実験結果

4.3.1　生成された質問の品質検証結果

品質検証の結果を表 1 に示す。現実性の評価において、67.5%の適合率が得られた。

4.3.2　質問タイプ制御の検証結果

質問タイプの分類結果を表 2 に示す。質問タイプの検証の結果、940 件すべての質問が Why 型に分類された。これにより、提案手法が Why 型質問を 100%の精度で生成できることが確認された。

5　考察

実験の結果、質問タイプの制御については、すべての質問が Why 型として正しく生成できた。これは、プロンプトにおいて Why 型の推奨フォーマットを明示し、What 型の形式（「〜の原因は何か？」「〜の理由は何か？」）を明示的に禁止したことが有効に機能したためと考えられる。

一方で、現実性の評価は 67.5%と、必ずしも高い値とはなっていない。現実性を 0 と判定した QA ペアの質問 Q を詳細に分析した結果、以下の 3 つの失敗類型が観察された。1 つ目の類型は、質問文への原因情報、すなわち、回答情報の混入である。これは、原因を問う質問をする時点では知らないはずの情報が質問文に記載されているケースであり、例えば「なぜ、加湿空調を採用した工場の 2 階天井面から屋根裏にリークした室内空気が結露し、火災報知器の誤作動につながったのか？」という質問では、「2 階天井面から屋根裏にリークした室内空気が結露したこと」自体が事故の原因であり、質問文に含まれるべきではない。2 つ目の類型は、事例概要に無い情報の混入である。LLM が質問生成時に、事例概要や失敗知識データベースの他の項目にも記載が無い情報を勝手に付け加えており、「東京都品川区の構造試験室」「2025 年 6 月 10 日」といった架空の具体的情報が質問文に挿入されるケースが確認された。3 つ目の類型は、事例概要に記載されていない情報への質問である。例えば、不具合の結果、ボルト 1 本から 2 本への変更という対策事例において、「なぜ以前はボルト 1 本で採用されていたのか」という事故事例発生前の設計理由を問う質問が生成されるなど、事例概要の内容とは異なる観点からの質問が生成されていた。

上記の実験結果より得られた課題を解決し、質問 Q の品質を向上させるため、今後は回答 A と質問 Q から質問 Q の妥当性を判定する LLM ベースのフィルタリング機構を実装する予定である。具体的には、質問文に原因情報が含まれていないか、質問文に事例概要に存在しない架空の情報が含まれていないか、質問の内容が事例概要から回答可能な範囲に収まっているかの 3 つの観点での検証を行うフィルタを設計する。このフィルタリング機構により、現実性を 0 と判定される質問を自動的に除外し、データセット全体の品質向上を図る。

6　おわりに

本稿では、失敗知識データベースを活用した Why 型 QA データセットの自動生成手法を提案した。提案手法は、Sentence-BERT による意味的類似度計算を用いた原因文の特定、LLM による Why 型質問の生成、QA ペアの構築という三段階のプロセスから構成される。実験の結果、提案手法により生成された質問すべてが Why 型として正しく分類されたものの、現実性の評価は 67.5%にとどまり、質問文への原因情報の混入、事例概要に無い架空情報の混入、事例概要から回答不可能な質問の生成という 3 つの失敗類型が明らかになった。現実性を向上させるため、3 つの失敗類型をフィルタリングをする機構の実装が今後の課題である。

参考文献

[1] Patrick 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. NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems, Pages 9459 - 9474, 2020.

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2024.

[3] 寺井孝則, 田原英一, 大石悠河, 湯浅晃. 実用的な品質観点に基づく RAG 性能評価用 QA データセットの自動生成. 言語処理学会 第 31 回年次大会 発表論文集, 2025 年 3 月.

[4] Caiming Xiong and Chien-Sheng Wu. Do RAG systems cover what matters? Evaluating and optimizing responses with sub-question coverage. In Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 5836–5849, 2025.

[5] 畑村洋太郎, 中尾政之, 飯野謙次. 失敗知識データベース構築の試み. 情報処理,Vol.44, No. 7, pp. 733–739, 2003.

[6] 失敗知識データベース, （2025-09-16 閲覧）. http://www.shippai.org/fkd/index.php.

[7] Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Ethan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Wen tau Yih, and Xin Luna Dong. CRAG – comprehensive rag benchmark. NIPS '24: Proceedings of the 38th International Conference on Neural Information Processing Systems, pages 10470 - 10490, 2024.