論文

Meta-Retrieversを用いたRAGの完全性破綻点の分析手法の提案

Proposal of a Method for Analyzing Integrity Failure Points of RAGs Using Meta-Retrievers

我妻 翔 WAGATSUMA Sho
デジタルハリウッド大学大学院 院生
Digital Hollywood University, Graduate School

白井 暁彦 SHIRAI Akihiko
デジタルハリウッド大学 特任教授
Digital Hollywood University, Project Professor

現代のLarge Language Models (LLM)を用いたチャットボット開発に、コンテキストの理解や長期記憶の実装は不可欠である。これらの実装にあたってLLMの外部記憶にRetrieval-Augmented Generation (RAG) は有用であると言われるが、オープンエンドな会話ではクエリとコーパスの整合性が合わないことが多い。Retrieverはコサイン類似度やスコアなどの数値を示す一方で、クエリに対してRetrieverが取得してきた情報の正しさは保証されないため、多様な方法が提案されている。本論文ではRetrieverの性能を評価するにあたってLLMがクエリとコーパスを動的に作成するMeta-Retrieversを提案し、Retrieverの完全性破綻点を特定・評価する分析手法の提案を試みた。実験結果として４つのRetrieverのうち３つのRetrieverにおいて完全性破綻点が存在することを確認した。本研究の成果として、LLMがクエリとコーパスを動的に作成することによって各種チャットボットの利用環境に近い状況を作り出し、クエリとコーパスに恣意的な要素を排除して多様な議論や用途のスコープによらず、メタ視点でモデルを評価することに成功している。これにより、Retriever単体の完全性破綻率をパーセンテージで示すことによってRAGシステムの初期設計段階における意思決定プロセスを支援し、より効率的なシステム開発につながる可能性を示している。

1. 導入

1.1 背景

著者らはこれまで、LLMを活用したエンタテイメントbotや確定申告等に利用できる実用的なサービスを開発してきた[1,2,3]。コンテキストの理解や長期記憶の実装が、ユーザー体験や価値設計の上で重要であることを確認してきたが、一方で技術実装的視点で近年、CSおよびIT分野で提案される、RAGに関する多くの論文等で提案される手法に疑問を持つようになった。それは、「RAGの評価に一般のユーザーを使ったオープンエンドな会話実験を行っていない」という点である。オープンエンドな会話、すなわち雑談や「質問者が回答を持ち得ていない質問」をRAGの評価データセットとして考えると、OpenAI Evals[4]のようなクエリとコーパスのセットを集合知的にオープンソースで集める手法も有用であるが、特定目的における恣意もしくはゴール設定が可能であるという意味でオープンエンドな評価であるとは言い難い。処理性能ランキングで１位を取るといった目的ではなく、実用的かつ実際のユーザーとのやり取りを想定してRAGを用いたチャットボットを評価する目的や、開発プロセスに組み込むことを想定すると、プロプライエタリな用途に向けた開発や概念実証 (Proof of Concept:PoC) のような初期段階でのモデル評価が重要になるため、動的かつ制御性高く生成できる手法があれば望ましいと考えた。実際のサービスに活かすためのRAG Retrieverの評価実験は、静的なデータセットではなく、動的なクエリとコーパスを用いて制御性高く評価できるべきではないだろうか。

RAGに限らず、ベクトル検索で使われるコサイン類似度は、単語の意味の距離を測り、完全一致ではなくても近い意味の文章を取得するための手法として役立つ。正規化手法にも依存するが、コサイン類似度はエラーを含めた文章の評価と捉えることもできる。例えば、コサイン類似度の１を完全一致と定義することは簡単であるが、0.9は極めて近い数値であるが0.1相違であることを示す。一方で、0や-1といった値が持つ意味は、「全く関係がない」か「非常に正解に近い偽(false)」である可能性もある。自然言語による雑談ではなく、現在LLMとして手軽に利用できるようになった推論ベースのテキスト生成を「真偽値や対偶といった論理的評価でシンプルに扱いたい」という課題は既存業務の置き換えに多く存在する。例えば税務のようなビジネス用途、ゲームシステムなどルールベースで判定すべきユーザー体験として多く残っている。このような用途では真偽の距離や対偶といった推論や確率的処理における類似度だけでなく、完全性、すなわち「真として判定する閾値をどのような値に設定すべきか」というメタ設計要素が残る問題を浮き彫りにしている。

近年の自然言語処理分野の国際研究では、Okapi BM25(以下BM25)などの高度なキーワード検索アルゴリズムは、単純な単語の一致以上の機能を提供する。BM25は文書長や単語の出現頻度、逆文書頻度 (IDF) を考慮し、より関連性の高い検索結果を提供できる。しかし、キーワード検索の基本的な限界として、文脈や意味的な関連性を十分に理解することができない。すなわち、単語の存在や頻度は評価できるが、その単語がどのような文脈で使用されているかを判断することは難しい。そのため、BM25のような高度なアルゴリズムを用いても、完全な情報取得や意味的に関連する全ての情報の抽出は困難である。

本研究で提案するMeta-Retrievers(以下MRAG)とは、RAGを用いたチャットボットによる対話的回答システムのワンショットでの動的評価を多数行い「完全性の破綻」の存在確認を行う。人力な静的評価では不可能な複数のクエリとコーパスを動的に作成し、設計評価時点で正答を判定するための閾値を評価可能にし、複数のRetrieverの性能評価を行うことを制御性高く実施できるツールを提案する。

本研究の調査対象で定義する「完全性破綻点」とは、コンテキストと関連性の高い情報が検索対象のコーパスの中に含まれているにもかかわらず、Retrieverがそれよりもコンテキストと関連性の低いコーパスに対して、コサイン類似度またはBM25のスコアで最も高い数値を算出した場合に完全性破綻点が存在すると定義する。

1.2 研究の目的

本研究の目的は以下である。キーワード検索およびベクトル検索の単独使用における完全性破綻点を実験的に検証し、それぞれのRetrieverの完全性破綻点が特定できる条件を明らかにする。また、動的に生成されるクエリとコーパスを用いた新しい評価手法を通じ

て、より実際の使用環境に近い条件下での各検索モデルの性能を比較分析する。

これらの知見に基づき、RAGシステムにおける最適な検索戦略の選択基準を確立し、システムの特性やデータの性質に応じた効果的なRAG実装のための指針を提示する。

2. 関連研究

2.1 RAGAS

RAGの性能評価は、人力以外にRAGASなどの評価ツールを用いる方法がある。RAGASは主に、RAGのFaithfulness(回答の忠実性)、Answer Relevance(回答の関連性)、Context Relevance(コンテキストの関連性)を評価する。回答の忠実性とは、生成された回答が与えられたコンテキストに基づいているか測るものである。回答の関連性とは、生成された回答がクエリに適切に答えているかどうかを測るものである。コンテキストの関連性とは、取得したコンテキストがどれだけ質問に焦点を当て、無関係な情報が含まれていないかを測るものである[5]。

本研究においてはRAGに用いるRetrieverの性能評価としてMRAGを提案する。

2.2 チャットボット開発における汎用的長期記憶の実装

これまで著者の開発した長期記憶を有するLINEチャットボットに「クッキングママ・ローラ」がある[1]。DynamoDBによる長期記憶を実装しており、最新5件の会話履歴とFunction Calling（以下FC）によって日付指定で過去の料理レシピ情報を取得してくる機能がある。

著者は他に2023年11月11日の「技術書典15」で発表した技術文書において「パーソナルトレーナBOTシンディ」[2]というAWS Lambdaで実装した長期記憶を有するLINEチャットボットも開発している。OpenAI GPT-3.5-turbo-0613におけるFCを用いてユーザーの発言から会話内容を推測してパーソナルデータを保存し、必要に応じて取得する試みがされているが、FCの誤りなど情報の保存と取得の仕組みにゆらぎや不完全な要素も多く、ユーザークエリの中から必要な情報を保存して取得する上では制御性が低く、実装の難しさが指摘されている。

さらに著者らが2023年9月頃から構想を練り、我妻が2024年1月から2024年3月まで開発に携わった「AI確定申告さん」(AI tax refund @ChaTax)[3]がある。現代的なLLMを活用した自由会話によるオープンエンドな雑談による対話を許容したAI確定申告さんの「茶托税子」（以下ChaTax）は、ユーザーからの問い合わせがきた際にベクトル検索を用いてコーパスから確定申告に関しての情報を取得する仕組みを構築した。ChaTaxにおいてユーザークエリは自由なので曖昧な単語の使用や省略された表現などもあり、1ショットの会話でRAGが取得してくる情報にコンテキストの関連性の高くない情報が含まれる場面がユーザーの実際のオープンエンド会話で確認されている。

国税に関する話題を取り扱うチャットボットの先行事例として国税庁のチャットボット「ふたば」[6]がある。「ふたば」はルールベースによってあらかじめ準備された回答テキスト群から類似度の近い回答を取得していると考えられる。ルールベースによるチャットボットは自由会話や質問の多様さには対応できないが、国税や確定申告に関することなどは回答群が有限であり、ユーザーが何の目的で情報を取得したいか回答する上での「完璧な正答」を定義できる。

ChaTaxと、既存の知識ベース、ルールベースのチャットボット「ふたば」（国税庁）の評価において優位性を認めることが難しいというパラドックスが起きることを体験した。ユーザー体験として「ふたば」は、必ず定型文としてのルールベースの処理に持ち込まれるため「ああ、これはボットなんだな」という「諦め」をユーザー体験に強いる一方で、LLMによるChaTaxはユーザーに寄り添ったコンテキストで多様な作文や判定を行うことができる。一方で「はい/いいえ」や「違法/適法」といった真偽値で判定すべき回答に対してワンショットのFCによる判定では制御性が非常に脆弱であることがわかった。キャラクターを活用したチャットボット開発や、ビジネス用途でのチャットボット開発は今後も需要があり、またユーザーはこの脆弱な制御性を利用して、判定難度が高い質問や、あえてコンテキストの理解を使った質問（例「昨日のあれだけど、大丈夫？」）を投じる傾向があるため、「システムとしての完全性破綻点をどこに設定するか」を制御することは、今後の研究課題として重要なポテンシャルを有すると考える。

2.3 ChatGPTでの業務効率化と断念

香川県三豊市の取り組みで、「ごみ出し案内」業務にChatGPTを活用しないと決断した事例がある。実証実験を始めた経緯を記事から引用すると、「市のWebサイトにもごみ出し案内の情報は掲載しているが、質問したい内容を探す必要があった。ChatGPTを使えば、知りたい情報にすぐ回答できると思った」とされており、「ごみ出し案内」業務のAI導入は2023年6月から実証実験を開始し、10月23日からはLLMのモデルをGPT-3.5からGPT-4に変更するなどして正答率の向上を図り、正答率94.1%まで向上したが三豊市の本格導入の条件は正答率99%としていたため、ごみ出し案内へのChatGPT導入は見送られることとなった、という経緯であったと報告されている[7]。ごみ出し案内などの正確性が求められる業務については、ルールベースを採用した方がエラーのない回答を提供することができるだろう。

三豊市が求めていたAIのクオリティに関する発言を記事から引用すると、「AIには少なくとも職員と同等のレベルを求め、それに達しない限り対市民向けとしては導入できないと考えた。また、AIがどのように回答したかを結局のところ職員が確認する作業が伴い、正答率が低ければそれだけ確認する頻度も上げなければならない。100%は無理としても99%は譲れない条件だった[8]」

人間である職員の対応は、ルールベースのシステムとは違い多様な質問への対応や行間を読む能力などの豊かなコミュニケーション力が期待される。正確性において厳格なルールベースを代替するAIに期待される能力は、人間の持つ豊かなコミュニケーション力に近いまたは同等の水準であると考えられるが、それに伴いルールベースの持つ正確性で妥協する場合、その許容範囲がエラー1%であるならば、採用するモデルの「完全性破綻点」を設計パラメータに入れて「完全性破綻率」を指標にするべきである。

2.4 Blended RAG

Dence Vector IndexやSparse Encoder Indexのようなセマンティック検索技術を、ハイブリッドクエリ戦略と融合させたBlend RAGによってRAGの精度が高まることがわかっている[9]。

しかし、どのようなシステムにおいても必ず複数の検索手法を用いる必要があるかどうか検討の余地がある。また、Blended RAGによって精度が上がったとしても使用するRetrieverに完全性破綻点が存在するのであればエラー発生の可能性は常に存在すると考える。

2.5 TF-IDF

用語頻度（TF）は、ある単語が文書中に何回出現するかを測定する指標である。例えば、5000語ある文書で"IT"が10回出現する場合、TFは次のように計算される。

TF= 10/5000 = 0.002

逆文書頻度（IDF）は、文書全体における単語の重要度を測る指標である。例えば、"ビジネス"という単語が2000文書中で10文書に存在する場合、IDFは次のように計算される。

IDF = log_e(10/5) = 0.3010

TF-IDFは、TFとIDFを掛け合わせた指標で、特定の文書内での単語の重要度を測定する。例えば、TFが0.002でIDFが0.3010の場合、TF-IDFは次のように計算される[10]。

TF-IDF = 0.0020.3010 = 0.000602

BM25はTF-IDFのアップグレードで具体的には、用語頻度の飽和と文書長を考慮することで、より精密な評価を行う[11]。

一方で、ベクトル検索はTF-IDFとは異なるアプローチを取る。TF-IDFは各単語の出現頻度と文書全体での出現頻度に基づいて単語の重要度を評価するが、ベクトル検索は単語や文書をEmbeddingしてベクトルとして表現し、そのベクトル間の距離や類似度に基づいて検索を行う。この手法は、単語の意味的な関係をよりよく捉えることができ、検索の精度を向上させる。ベクトル検索は文書全体における単語の影響を測定せず、1つのクエリに対して1つのコーパスのコサイン類似度を測る。

3. 提案手法

3.1 MRAGの提案

本研究では、著者が自作した2つのRetriever評価ツール(MRAG[12])を用いてRetrieverに完全性破綻点が存在するか実験し、存在した場合に「完全性破綻率」をエラー回数/対象のRetrieverの試行回数によりパーセンテージで算出する。

データセットは著者がPythonで作成した2つのMRAG[12]を用いて実験をする。

1つ目は、「Six Types Of Proper Noun Meta-Retrievers」[12]で、入力されたエンティティに基づいてLLMが動的にクエリを作成する。エンティティは、ポピュラーな人物名を入力する。クエリの内容は、そのエンティティ(人物)に関する業績などの質問となる。一度に最大6つの人物名を入力できる。それぞれのクエリに対応した回答を動的に生成して、それらをコーパスとしてデータセットを構築する。

2つ目は、「Six-Dimensional Query Meta-RetrieversI」[12]でSix Types Of Proper Noun Meta-Retrieversとの差分は、入力された1つのエンティティに基づいて6つのクエリを作成する点である。

クエリは質問内容が重複しないように6つのフォーマットに従って人物に関する質問を生成する。

6つのフォーマットはシステムプロンプトであり、実際に使われているシステムプロンプトを和訳すると、「出生と家庭環境」、「教育と訓練経験」、「初期キャリアと貢献」、「主要業績と評価」、「プライベートと性格」、「影響と文化的インパクト」についての質問が生成されるようにプロンプトエンジニアリングとFCを用いて作成した。

例えば「Napoleon Bonaparte」(ナポレオン・ボナパルト)という人物を入力すると、その人物に関する6つの質問が生成される。実際に生成されたナポレオンの「出生と家庭環境」に関するクエリは、

“How did Napoléon Bonaparte's Corsican heritage and family background influence his early identity and ambitions?”

「ナポレオン・ボナパルトのコルシカ島の出自と家庭環境は、彼の初期のアイデンティティと野心にどのような影響を与えたのだろうか?」

上記のクエリに対して、コンテキストの関連性の高いコーパスをLLMが生成する。実際に生成された回答は、

“Napoléon Bonaparte was born in 1769 in Corsica, shortly after the island was annexed by France. His Corsican heritage instilled a strong sense of identity and pride in local culture, shaping his early worldview. His family, of minor nobility, emphasized education and ambition, which encouraged his aspirations for military and political success. Napoleon's exposure to the island's struggle for independence fostered a desire for power and authority, motivating him to rise through the ranks of the French military”

「ナポレオン・ボナパルトは1769年、コルシカ島がフランスに併合された直後にコルシカ島で生まれた。コルシカ人の血を引く彼は、強いアイデンティティ意識と地元文化への誇りを植え付けられ、初期の世界観を形成した。彼の家系は下級貴族で、教育と野心を重視し、それが彼の軍事的、政治的成功への願望を後押しした。ナポレオンは島の独立闘争に触れることで、権力と権威への欲望を育み、フランス軍での出世の動機となった」

上記のクエリとコーパスの組み合わせは出力されるJSONファイルの「query_1」と「answer_1」のキーのバリューとなる。

クエリは6つ生成されるので、query_6とanswer_6まで生成される。queryキーとanswerキーの末尾の数字が同じバリュー同士が、MRAGが定義する最もコンテキストの関連性の高いクエリとコーパスの組み合わせである。

maxTokensの指定ができてmaxTokensの80%の数値がminTokensとなる。minTokensに満たないTokens数の回答が生成された場合、拡張回答を生成する関数を呼び出してminTokensを超える規模に拡張する。

3.2 評価基準とMRAGの使用理由

MRAGのエラーの条件は、queryキーとanswerキーの末尾の数字が違う数字のバリューのコサイン類似度またはスコアの数値が高かった場合にエラー1とする。

MRAGは、LLMが動的にクエリとコーパスを作成する。これによって、より実際のサービスの自由会話に近い環境を作り出し恣意的な要素を排除してRetrieverの性能実験を行う。作成したクエリやコーパスが、LLMのハルシネーションによって間違った情報を作り出している可能性は存在する。

しかし、オープンエンドな会話の会話履歴をコーパスにする場合において、そのコーパスの情報が正しいか正しくないかによってRAGの情報取得が期待されるわけではない。取得した情報が会話履歴であるならば、会話の内容自体が正しいか正しくないかではなく、会話のコンテキストが成立することが最も重要である。

そのためにLLMの性能として、回答の忠実性と回答の関連性が求められ、Retrieverの性能としてコンテキストの関連性が求められる。

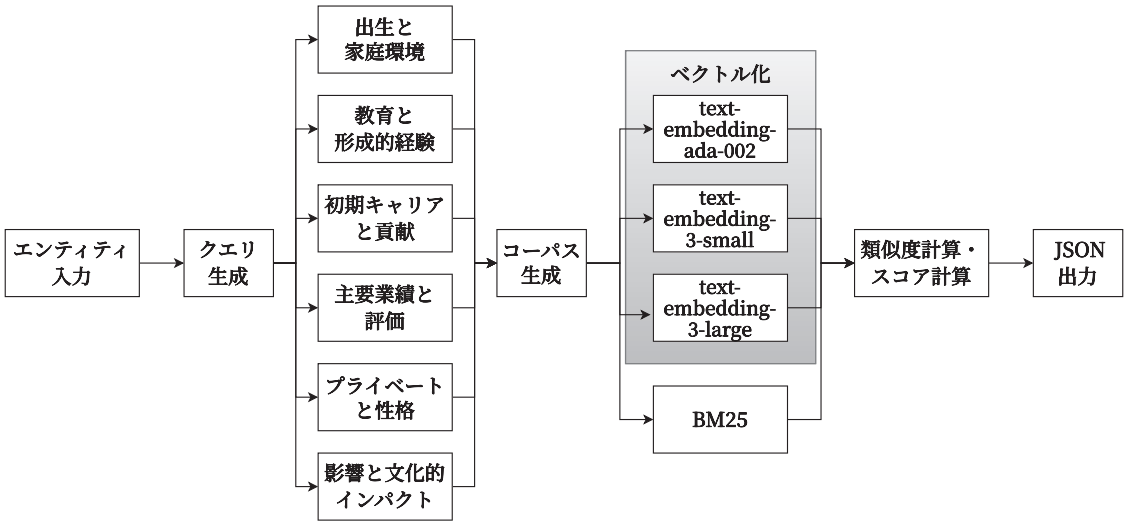

図1 Six-Dimensional Query Meta-Retrieversのアプリケーションフロー図

3.3 使用モデルと出力結果

LLMはOpenAIのgpt-4o mini-2024-07-18を使用。

MRAGのtemperatureはそれぞれ0.0、0.5、0.7、0.9に設定する。seedは42に固定する。

Retrieverのモデルとしてtext-embedding-3-largeとtext-embedding-3-smallとtext-embedding-ada-002を使用する。キーワード検索はBM25を使用。

エンティティとtemperatureとmaxTokenを選択してSubmitボタンを押すと、クエリとコーパスに対して４つのモデルの処理が走る。

出力結果としてJSONファイルが出力される。

ファイル名の構成は、「ファイルの種類_ミリ秒までの日付_maxTokens_temperature」が記載されていて、その条件の試験結果が出力される。

ファイルの種類が、「best_matches」のファイルには、コサイン類似度またはスコアが最も高かったクエリとコーパスの組み合わせが記載される。「cosine_similarities」または「scores」のファイルは、クエリに対してのコーパスのコサイン類似度またはスコアの数値が全て記載される。「vectors」は試験に使用するテキストの埋め込みベクトル、「queries」と「corpus」はそれぞれ生成したクエリとコーパスが記載されている。

4. 実験

4.1 定義

エラーを１度でも確認したRetrieverには限定的な条件下であるが、完全性破綻点が存在すると定義する。

限定的な条件は使用するMRAGによって定義が異なる。

Six Types Of Proper Noun Meta-Retrieversの実験でRetrieverにエラーが確認された場合、対象のRetrieverはコーパス全体の主語となる固有名詞が別離された場合であっても、完全性破綻点が発生する可能性があると定義する。

Six-Dimensional Query Meta-Retrieversの実験でRetrieverにエラーが確認された場合、コーパス全体の主語となる固有名詞が同じだが、その固有名詞に関する話題が別離された場合であっても完全性破綻点が発生する可能性があると定義する。

4.2 試行回数の計算方法

試行回数は対象の１つのクエリに対して、１つのコーパスのコサイン類似度またはスコアを測定した時点で１回とする。maxToken100-575の試験を行い、25token刻みで増やしていく。１つのエンティティに対してMRAGは６つのクエリと６つのコーパスを作成する。入力するエンティティは４パターンを試す。１つのRetrieverのmaxToken100の試行回数はエンティティが４つなのでクエリ６個をかけて24、temperatureのパターンが0.0、0.5、0.7、0.9で４つあるので４をかけて96回。最大575Tokenまで25token刻みで増やすので20をかけて合計1920回。対象のRetrieverは４つあるので7680回。Meta-RetrieversはSix Types Of Proper Noun Meta-RetrieversとSix-Dimensional Query Meta-Retrievers の２つあるので15360回試行した。

4.3 実験１ Six Types Of Proper Noun Meta-Retrieversの実験

表1 Six Types Of Proper Noun Meta-Retrieversの実験結果

Error Count					
Models/Max Tokens	100-200	225-325	350-450	475-575	Total
text-embedding-3-large	0 /480	0 /480	0 /480	0 /480	0 /1920
text-embedding-3-small	0 /480	0 /480	0 /480	0 /480	0 /1920
text-embedding-ada-002	0 /480	0 /480	1 /480	0 /480	1 /1920
BM25	0 /480	1 /480	1 /480	1 /480	3 /1920
Total	0 /1920	1 /1920	2 /1920	1 /1920	4 /7680
Error Rate					
Models/Max Tokens	100-200	225-325	350-450	475-575	Total
text-embedding-3-large	0.00%	0.00%	0.00%	0.00%	0.00%
text-embedding-3-small	0.00%	0.00%	0.00%	0.00%	0.00%
text-embedding-ada-002	0.00%	0.00%	0.21%	0.00%	0.05%
BM25	0.00%	0.21%	0.21%	0.21%	0.16%
Total	0.00%	0.05%	0.10%	0.05%	0.05%

全ての実験結果はこちらで参照可能（参照2024年8月31日）[13]。実験結果に基づいたエラーカウントの分析結果はこちらで参照可能（参照2024年8月31日）[14]。

4.4 実験２ Six-Dimensional Query Meta-Retrieversの実験

表2 Six-Dimensional Query Meta-Retrieversの実験結果

Error Count					
Models/Max Tokens	100-200	225-325	350-450	475-575	Total
text-embedding-3-large	0 /480	0 /480	0 /480	0 /480	0 /1920
text-embedding-3-small	1 /480	0 /480	0 /480	0 /480	1 /1920
text-embedding-ada-002	0 /480	0 /480	0 /480	0 /480	0 /1920
BM25	3 /480	2 /480	2 /480	3 /480	10 /1920
Total	4 /1920	2 /1920	2 /1920	3 /1920	11 /7680
Error Rate					
Models/Max Tokens	100-200	225-325	350-450	475-575	Total
text-embedding-3-large	0.00%	0.00%	0.00%	0.00%	0.00%
text-embedding-3-small	0.21%	0.00%	0.00%	0.00%	0.05%
text-embedding-ada-002	0.00%	0.00%	0.00%	0.00%	0.00%
BM25	0.63%	0.42%	0.42%	0.63%	0.52%
Total	0.21%	0.10%	0.10%	0.16%	0.14%

全ての実験結果はこちらで参照可能(参照2024年8月31日)[15]。実験結果に基づいたエラーカウントの分析結果はこちらで参照可能(参照2024年8月31日)[16]。

5. 結果

5.1.1 Six Types Of Proper Noun Meta-Retrieversでの完全性破綻点の存在の確認

text-embedding-ada-002に1つ、BM25に3つのエラーを確認した。text-embedding-ada-002でエラーが発生した条件は入力エンティティが"Michael Jackson"、"Whitney Houston"、"David Bowie"、"Freddie Mercury"、"Ariana Grande"、"Bob Marley"、maxTokens 425、temperature 0.0。出力結果はこちらで参照可能(参照2024年8月31日)[17]。

出力結果のJSONファイルの、「best_matches_20240829_190704_425_0.0.json」を確認すると、query_5のbest_answer_keyのバリューがanswer_2となっている。query_5のquery_textのバリューは、

"query_text": "How has Ariana Grande's blend of pop and R&B, along with her vocal range, shaped contemporary music trends and influenced a new generation of artists?"

「アリアナ・グランデのポップとR&Bの融合とその歌唱力は、どのように現代の音楽トレンドを形成し、新世代のアーティストに影響を与えたか?」

best_answer_text(answer_2)のバリューは、

“Whitney Houston's vocal techniques and emotional delivery significantly influenced the standards of pop and R&B music during her career, particularly in the 1980s and 1990s. Her powerful, technically proficient voice set a new benchmark for vocal performance in these genres…(以下省略)”

「ホイットニー・ヒューストンのヴォーカル・テクニックとエモーショナルな歌唱は、彼女のキャリア、特に1980年代と1990年代のポップスとR&Bのスタンダードに大きな影響を与えた。彼女のパワフルで技術的に熟達した歌声は、これらのジャンルにおけるヴォーカル・パフォーマンスの新たな基準を打ち立てた…(以下省略)」

Tokensが長大なのでここでは省略させていただくが、アリアナ・グランデに関するクエリに対してホイットニー・ヒューストンのコーパスのコサイン類似度を最も高く設定している。query_5のコサイン類似度の数値はそれぞれ、 "answer_1":0.758,"answer_2":0.850, "answer_3":0.795,"answer_4":0.769,"answer_5":0.843, "answer_6": 0.761。、となっている(小数点第3位以下は切り捨てで表記)。

answer_5のコーパスの内容は、

“Ariana Grande's blend of pop and R&B, characterized by her impressive vocal range and emotive delivery, has significantly shaped contemporary music trends since her rise to fame in the early 2010s. Her ability to seamlessly merge these genres has not only broadened the appeal of R&B-infused pop but has also redefined the sound of mainstream music…(以下省略)”

「アリアナ・グランデのポップとR&Bの融合は、彼女の印象的なヴォーカル・レンジと感情的な表現によって特徴付けられ、2010年代初頭に有名になって以来、現代の音楽のトレンドを大きく形成してきた。これらのジャンルをシームレスに融合させる彼女の能力は、R&Bを取り入れたポップの魅力を広げただけでなく、メインストリーム・ミュージックのサウンドを再定義した…(以下省略)」

LLMが正答と位置付けたanswer_5の数値の高さは上から2番目になっているため、完全性破綻点を確認したと言える。

text-embedding-ada-002において、コーパス全体の主語となる固有名詞が別離された場合であっても完全性破綻点の存在を確認することができた。完全性破綻率は0.05%。

次に、BM25は3つのエラーを確認している。

エラーの例を挙げると入力エンティティは"Taylor Swift"、"Ed Sheeran"、"Adele Laurie Blue Adkins"、"Beyoncé Knowles Carter"、"Bruno Mars"、"Lady Gaga"、maxTokens 325、temperature 0.7。出力結果はこちらで参照可能(参照2024年8月31日)[18]。

出力結果のJSONファイルの、「best_matches_20240830_003953_325_0.7.json」を確認するとquery_1のbest_answer_keyのバリューがanswer_3となっている。query_1のquery_textのバリューは、

“How has Taylor Swift's songwriting evolved over the years, particularly in terms of themes and musical styles, as evidenced by her albums from 'Taylor Swift' to 'Midnights'?”

「テイラー・スウィフトのソングライティングは、『テイラー・スウィフト』から『ミッドナイツ』までのアルバムに見られるように、特にテーマや音楽的スタイルにおいて、長年にわたってどのように進化してきたか?」

best_answer_text(answer_3)のバリューは、

“Adele Laurie Blue Adkins, known simply as Adele, achieved remarkable success in the music industry with her albums '21' and '25' due in large part to her distinctive vocal style and profound emotional depth. Released in 2011, '21' showcased her powerful contralto voice, which is characterized by its rich tone and impressive range…(以下省略)”

「アデルとして知られるアデル・ローリー・ブルー・アドキンスは、独特のヴォーカル、スタイルと深遠なる感情の深さにより、アルバム『21』と『25』で音楽業界で目覚ましい成功を収めた。 2011年にリリースされた『21』は、豊かな音色と印象的な音域を特徴とする彼女のパワフルなコントラルト・ヴォイスを披露した…(以下省略)」

query_1のスコアはそれぞれ"answer_1":6.264, "answer_2":6.110,"answer_3":7.305,"answer_4":4.353,"answer_5":6.613, "answer_6": 4.760、となっている。

answer_1のコーパスの内容は、

“Taylor Swift's songwriting has undergone significant evolution from her self-titled debut album "Taylor Swift" (2006) to "Midnights" (2022), reflecting changes in both themes and musical styles. In her early work…(以下省略)”

「テイラー・スウィフトの曲作りは、セルフタイトルのデビューアルバム『Taylor Swift』(2006年)から『Midnights』(2022年)まで、テーマと音楽スタイル両方の変化を反映しながら大きな進化を遂げてきた。初期の作品では…(以下省略)」

内容を見ても、明らかにLLMが正答と位置付けたanswer_1のコーパスの方がクエリに対してコンテキストの関連性が高い。

これはTF-IDFが単語の出現頻度と、コーパス全体の中での単語の重要度を測るというアルゴリズムであるため、入力したエンティティ以外の共通した単語と他コーパスであまり出現しなかった単語によってスコアが高まったと考えられる。ベクトル検索がクエリとコーパスのコサイン類似度を測定するとき、他のコーパスの影響を受けないのに対して、BM25はコーパス全体の中での単語の重要度も測る。

クエリとコーパスの単語の出現頻度を調べてみると、query_1に含まれる「albums」という単語はanswer_3にだけ含まれていて3つあった。BM25の文書長の考慮は、どのコーパスも大体同じ長さになるようにmaxTokensの長さから80%をminTokensに設定しているため影響は小さいと考えられる。BM25はコーパス全体の主語となる固有名詞が別離された場合であっても、クエリの中に含まれる単語がコーパスの中に出現し、そのコーパスが他のコーパスに含まれない場合に、コンテキストの関連性が低いまたは一切関連

性のないコーパスのスコアが最も高くなり、完全性破綻点を特定することができたといえる。完全性破綻率は0.16%。

5.1.2 Six-Dimensional Query Meta-Retrieversでの完全性破綻点の存在の確認

text-embedding-3-smallで1つ、BM25で10のエラーを確認した。text-embedding-3-smallでエラーが発生した条件は入力エンティティが"George Washington"、maxTokens 125、temperature 0.0。出力結果はこちらで参照可能（参照2024年8月31日）[19]。

出力結果のJSONファイルの、「best_matches_20240829_181956_125_0.0.json」を確認するとquery_6のbest_answer_keyのバリューがanswer_5となっている。query_6のquery_textのバリューは、

“In what ways has George Washington's leadership and vision impacted American political culture and identity, both during his time and in contemporary society?”

「ジョージ・ワシントンのリーダーシップとビジョンは、彼の時代と現代社会の両方において、アメリカの政治文化とアイデンティティにどのような影響を与えたか？」

best_answer_text(answer_5)のバリューは

“George Washington's personal relationships and character traits significantly influenced his decision-making and public image, particularly during the Revolutionary War and his presidency. His strong sense of duty and integrity earned him the trust of his peers and soldiers…(以下省略)”

「ジョージ・ワシントンの個人的な人間関係と性格的特徴は、特に独立戦争中と大統領在任中、彼の意思決定と公的イメージに大きな影響を与えた。彼の強い義務感と誠実さは、仲間や兵士たちからの信頼を得ていた（以下省略）」

query6のコサイン類似度の数値はそれぞれ、"answer_1":0.6079,"answer_2":0.511,"answer_3":0.571,"answer_4":0.6133,"answer_5":0.665,"answer_6": 0.662となっている。

query_6のコーパスの内容は

“George Washington's leadership during the American Revolution and as the first President established foundational principles of American governance, emphasizing unity, republicanism, and the importance of a strong federal government…(以下省略)”

「ジョージ・ワシントンはアメリカ独立戦争中、そして初代大統領としてリーダーシップを発揮し、団結、共和主義、強力な連邦政府の重要性を強調し、アメリカ統治の基本原則を確立した（以下省略）」

クエリの中にある「リーダーシップ」や「ビジョン」に関する質問が、ジョージ・ワシントンの個人的な人間関係と性格的特徴という文脈につながっていたため、answer_5のコサイン類似度の数値が高まったと考えられる。answer_5とanswer_6のコサイン類似度の数値はほぼ同じと言えるが、僅かにanswer_5が上回った。

text-embedding-3-smallにおいて、コーパス全体の主語となる固有名詞が同じだが、その固有名詞に関する話題が別離された場合であっても完全性破綻点を確認することができた。完全性破綻率は0.05%。

次に、BM25は10のエラーを確認している。

エラーの例を挙げると入力エンティティは”Winston Churchill”、maxTokens 100、temperature 0.9。出力結果はこちらで参照可能（参照2024年8月31日）[20]。

出力結果のJSONファイルの、「best_matches_20240830_012547_100_0.9.json」を確認するとquery_1のbest_answer_keyのバリューがanswer_3となっている。

query_1のquery_textのバリューは、

"How did Winston Churchill's family background and upbringing influence his views on leadership and politics during his early years?"

「ウィンストン・チャーチルの家庭環境や生い立ちは、彼のリーダーシップや政治に対する考え方にどのような影響を与えたか？」

best_answer_text(answer_3)のバリューは、

“Winston Churchill began his career in journalism and military service, participating in conflicts like the Second Boer War, where he gained fame as a war correspondent. His return to Britain marked his entry into politics, initially as a Conservative MP before switching to the Liberal Party. Key contributions included advocating for social reforms and naval expansion.…(以下省略)

「ウィンストン・チャーチルは、ジャーナリズムと軍務でキャリアをスタートさせ、第二次ボーア戦争などの紛争に参加し、特派員として名声を得た。英国に戻ると政界に進出し、当初は保守党議員として活躍したが、その後自由党に移った。社会改革や海軍の拡張を提唱するなど、重要な貢献を果たした。(以下省略)」

query_1のスコアはそれぞれ、"answer_1":4.983,"answer_2":4.653,"answer_3":6.544,"answer_4":2.6594,"answer_5":2.971,"answer_6": 2.572となっている。

answer_1のコーパスの内容は、

“Winston Churchill was born into an aristocratic family; his father, Lord Randolph Churchill, was a prominent Conservative politician, and his mother, Jennie Jerome, was an American socialite. This privileged upbringing immersed him in political discourse and the workings of British society from a young age. …(以下省略)”

「ウィンストン・チャーチルは貴族の家庭に生まれた。父ランドルフ・チャーチル卿は著名な保守党の政治家であり、母ジェニー・ジェロームはアメリカの社交界の華であった。このような恵まれた環境で育ったチャーチルは、若い頃から政治談義やイギリス社会の仕組みにどっぷりと浸かった。(以下省略)」

answer_1に比べてanswer_3は「出生と家庭環境」についての質問に答えていない。BM25においてコーパス全体の主語となる固有名詞が同じだが、その固有名詞に関する話題が別離された場合であっても、完全性破綻点を確認することができた。完全性破綻率は0.52%。

6. 考察

コーパス全体の主語となる固有名詞が別離された場合の、全Retrieverの完全性破綻率は0.05%、コーパス全体の主語となる固有名詞が同じだが、その固有名詞に関する話題が別離された場合の完全性破綻率は0.14%で2.8倍の差となる。

実際のサービスにおいても、コーパス全体の主語となる固有名詞は別離されていた方がRAGの精度が上がると考えられる。

コーパス全体の主語となる固有名詞が同じ場合でもBM25の完全性破綻率が0.52%と高く、話題は別離されているが含まれる単語の頻度やコーパス全体での出現頻度などで測定する場合、文脈を考慮していないためにクエリの中の質問内容全てに答えていないコーパスに最もスコアが高くなるなどの例があった[20]。

BM25は主に検索エンジンなどで採用されているが、検索エンジンのクエリは通常会話のような問いかけではなく「ナポレオン・ボナパルト」などのように単語を入力してサーチする目的で使われる。

ベクトル検索のように、Embeddingの必要がないため大量のデータを検索するのに向いているが、コーパスの規模が限られているRetrieverとして使う場合、クエリに無駄な単語が含まれないように条件を付与する必要がある。そのため、自由会話を楽しむbotのRAGシステムとしてBM25を使用する場合、ベクトル検索ほど性能を発揮できないことが予測できる。

今回の実験で、text-embedding-3-largeで完全性破綻点を確認することができなかったが、text-embedding-3-largeがコーパス全体の主語となる固有名詞が別離された場合と、コーパス全体の主語となる固有名詞が同じだがその固有名詞に関する話題が別離された場合に、絶対に完全性破綻点を確認することはないと断言することはできない。

しかし、他のRetrieverと比べてコンテキストの関連性をコサイン類似度を用いたスコアリングによって評価する性能が高いことはデータから観測できる事実である。

2024年8月31日 現 在、text-embedding-3-largeはOpanAIの提供するEmbeddingモデルの中で最もコストが高く最も性能が良いとされる[21]。最も性能が良いとされるモデルが、MRAGが正答と位置付けたクエリとコーパスの組み合わせと同じ組み合わせに最も高いコサイン類似度の数値を算出したことは、MRAGのRetriever評価ツールとしての手法が正しかったと考えられる。

7. 結論

本研究では、キーワード検索およびベクトル検索の単独使用における完全性破綻点を実験的に検証し、4つのRetrieverのうち3つのRetrieverの完全性破綻点が特定できる条件を明らかにした。本研究の主な価値は以下の3点にある。

1つ目は4つのRetrieverのうち3つのRetrieverの完全性破綻点と完全性破綻率を特定・分析できたこと。コーパス全体の主語となる固有名詞が別離された場合とコーパス全体の主語となる固有名詞が同じだが、その固有名詞に関する話題が別離された場合は、主語となる固有名詞が別離された場合の方がRetrieverによる差分はあるが完全性破綻率は低い傾向にあることが示された。コーパスの内容とサービスに求められるRAGの性能次第だが、Retriever単体で十分な性能を発揮できる条件はあると考えられる。また、いずれのRetrieverにおいても完全性破綻率は1%以下であり、期待される機能やサービスによるが、既存システムや職員などが遂行する人力の業務をRAGシステムが代替する可能性はあるといえる。

2つ目は、LLMを用いた動的クエリと動的コーパスによる実験で、恣意的な要素を排除してモデルを評価することができたこと。この成果によりLLMを用いることで被験者を集めて会話実験をするなどして人件費などのコストをかけることなく評価実験を行うことができた。複数の質問を生成する場合、内容を別離して質問を生成するためにFCなどの手段は有効である。

3つ目は、LLMを用いた動的クエリと動的コーパスを作成する際の注意点を発見したこと。結果に大きな差分が生まれるためプロンプトエンジニアリングによってルールや条件の詳細を設定する必要がある。かつ、コーパスの規模が長大になりすぎるとLLMが生成するコーパスの内容の崩壊を招く恐れがあり、実際のサービスにおいても、大きすぎるコーパスはチャンクに分割して使用されることが多いため、Tokensの規模は適正な範囲で設定する必要がある。

本研究の成果は、RAGシステムの初期設計段階における意思決定プロセスを支援し、より効率的なシステム開発につながる可能性があること。また、RAGを評価する手法やユーザークエリを必要とする研究は多いが、それらの実験においてLLMを用いればコストカットした上で恣意的な要素を排除してRetrieverを評価できることがわかった。

text-embedding-3-largeは本研究において、完全性破綻点を特定できなかったが、これは同Retrieverの高い性能を示唆している可能性がある。本研究の結果は限定的な条件下で得られたものであり、実際のサービスにおいては、複数の固有名詞が主語になっていたりもっと複雑になっていたりする場合も多い。そういった場合はBlended RAGなどのハイブリッド検索の手法の導入が有効であると考えられる。

今後の展望として、より多様な条件下での実験や実際のサービス環境に近い複雑なシナリオでの評価が必要である。また、text-embedding-3-largeなど精度の高いRetrieverをさらに詳細に分析することで、より確実性の高いRAGシステムの構築につながる可能性がある。特に厳格な正確性が求められる分野への適用に向けて、継続的な研究と改善が期待される。

本研究は、動的クエリと動的コーパスによる新しいRetrieverの評価手法を通じてRAGの課題を明らかにし、今後のチャットボットやサービスにおけるRAGシステムの発展に貢献する重要な一歩となった。これらの知見を基に、より効率的で信頼性の高いRAGシステムの開発が進むことが期待される。

参考文献

[1] WAGATSUMA Sho,IKEDA Akitoshi,HIRAMATSU Takeo, SHIRAI Akihiko:“GPTとクラウドを利用したチャットボット開発における汎用的長期記憶の実装”DHU JOURNAL Vol.10 2023 (December 2023), p2.
https://msl.dhw.ac.jp/wp-content/uploads/2023/12/DHUJOURNAL2023_P025.pdf

[2] ChatGPTとAWSで実装する長期記憶ボットの作り方
https://techbookfest.org/product/hp5REtamTy9QE5NvgjsMpP?productVariantID=nSq6V7Ygg3caiiZCkewYx9

[3] [Cha Tax]AI確定申告さん
https://corp.aicu.ai/chatax
Press released by AICU Inc. 19th Feb. 2024.

[4] OpenAI Evals, https://github.com/openai/evals, accessed on 31st, August, 2024.

[5] Shahul Es, Jithin James, Luis Espinosa-Anke, Steven Schockaert:”RAGAS: Automated Evaluation of Retrieval Augmented Generation”arXiv(26 Sep 2023), p3.
https://arxiv.org/abs/2309.15217

[6] 国税庁:チャットボット(ふたば)に質問する
https://www.nta.go.jp/taxes/shiraberu/chatbot/index.htm

[7] 松浦立樹,ITmedia NEWS:ChatGPTでの業務効率化を“断念”――正答率94%でも「ごみ出し案内」をAIに託せなかったワケ　三豊市と松尾研の半年間
https://www.itmedia.co.jp/news/articles/2312/15/news158.html (2023年12月15日 16時14分 公開)

[8] 松浦立樹,ITmedia NEWS:ChatGPTでの業務効率化を“断念”――正答率94%でも「ごみ出し案内」をAIに託せなかったワケ　三豊市と松尾研の半年間
https://www.itmedia.co.jp/news/articles/2312/15/news158_2.html(2023年12月15日 16時14分 公開)

[9] Kunal Sawarkar, Abhilasha Mangal, Shivam Raj Solanki:”Blended RAG:Improving RAG (Retriever-Augmented Generation) Accuracy with Semantic Search and Hybrid Query-Based Retrievers”arXiv(4 Aug 2024), p1.
https://arxiv.org/abs/2404.07220

[10] Shahzad Qaiser, Ramsha Ali:”Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents” ResearchGate(July 2018), p1.
https://www.researchgate.net/publication/326425709_Text_Mining_Use_of_TF-IDF_to_Examine_the_Relevance_of_Words_to_Documents

[11] TFIDF & BM25
https://guillim.github.io/datascience/2020/08/11/TFIDF-BM25.html (2020年8月11日)

[12] Meta-Retrievers
https://github.com/flymywife/Meta-Retrievers

[13] Six Types Of Proper Noun Meta-Retrieversの実験結果
https://github.com/flymywife/Meta-Retrievers/tree/main/

Results_SixTypesOfProperNounMeta-Retrievers
[14] Six Types Of Proper Noun Meta-Retrieversの実験結果分析
https://github.com/flymywife/Meta-Retrievers/tree/main/Analyze-SixTypesOfProperNounMeta-Retrievers
[15] Six-Dimensional Query Meta-Retrieversの実験結果
https://github.com/flymywife/Meta-Retrievers/tree/main/Results_Six-DimensionalQueryMeta-Retrievers
[16] Six Dimensional Query Meta-Retrieversの実験結果分析
https://github.com/flymywife/Meta-Retrievers/tree/main/Analyze_Six-DimensionalQueryMeta-Retrievers
[17] Six Types Of Proper Noun Meta-Retrieversのtext-embedding-ada-002エラーファイル
https://github.com/flymywife/Meta-Retrievers/blob/main/Results_SixTypesOfProperNounMeta-Retrievers/entities_Michael_Jackson_Whitney_Houston_David_Bowie_Freddie_Mercury_Ariana_Grande_Bob_Marley/text-embedding-ada-002/best_matches_20240829_190704_425_0.0.json
[18] Six Types Of Proper Noun Meta-RetrieversのBM25エラーファイル
https://github.com/flymywife/Meta-Retrievers/blob/main/Results_SixTypesOfProperNounMeta-Retrievers/entities_Taylor_Swift_Ed_Sheeran_Adele_Laurie_Blue_Adkins_Beyonc%C3%A9_Knowles_Carter_Bruno_Mars_Lady_Gaga/BM25/best_matches_20240830_003953_325_0.7.json
[19] Six-Dimensional Query Meta-Retrieversのtext-embedding-3-smallエラーファイル
https://github.com/flymywife/Meta-Retrievers//blob/main/Results_Six-DimensionalQueryMeta-Retrievers/historical_George_Washington/text-embedding-3-small/best_matches_20240829_181956_125_0.0.json
[20] Six-Dimensional Query Meta-RetrieversのBM25エラーファイル
https://github.com/flymywife/Meta-Retrievers/blob/main/Results_Six-DimensionalQueryMeta-Retrievers/historical_Winston_Churchill/BM25/best_matches_20240830_012547_100_0.9.json
[21] OpenAI Platform:Vector embedding
https://platform.openai.com/docs/guides/embeddings/embedding-models