ChatGPT

RAG教育コンテンツメニュー

データベース作成フェーズ

ドキュメント取り込み戦略

- **電子化する情報の範囲**: RAGの知識データベースとしてはテキストデータが中心になります。しかし、図や表などの**非テキスト情報**についても適切に意味を抽出・管理することが精度向上の重要なポイントです [1]。実際、**マルチモーダルRAG**ではテキスト以外に画像・音声・動画を統合して検索することも可能であり [2]、必要に応じてOCR等で画像からテキスト化するなど検討します。
- **メタデータ付与**: インデックス化する各ドキュメントには、**ファイル名、ページ番号、見出し構造、作成日**などのメタデータを付与することが不可欠です [3]。付与したメタデータは「検索での絞り込み → ランキングでのスコア向上 → 回答生成での再構成」の全段階に効き、検索精度や回答の一貫性を高める効果があります [4]。
- **文章の正規化**: Embeddingにかける前にテキストの**ノイズ除去や正規化**を行います。例えば**余計な空白の削除**や改行コードの統一、不要なHTMLタグや特殊記号の除去などでテキストを綺麗に整形します [5]。PDF由来の単語分断（ハイフネーション）の修復や、箇条書き記号の統一など細かな前処理も有用です [6]。
- **用語の統一**: 文書内外で表記ゆれがある専門用語については、**表記の揺れを統一**して一貫性を持たせます。例えば全角半角の統一や、同義語の統一的な置換を行うことで、情報の関連性が高まり検索精度が向上します [7]。実際に**正規化による表記ゆれの解消**はRAGにおける重要な前処理手法の一つです [7]。
- **他にはどんなものがある？**: その他にも、**重複データの排除**やフォーマットの統一などの前処理が考えられます。例えばデータを一律に**Markdown形式に変換**して構造を明示することで、AIが情報を理解しやすくなる効果があります [8]。これら前処理を徹底することでRAGの精度低下要因を事前に取り除き、後工程の負荷を減らすことができます。

エンベディングモデル選定戦略

RAGで使用する**ベクトル埋め込み（Embedding）モデルの選定**も精度に影響する重要ポイントです。ユーザ質問と文書の意味的な近さを数値化する処理はEmbeddingモデルによって精度が大きく左右されます [9]。そのため、用途に応じて**多言語対応**や**ドメイン特化**したモデル、高次元で精度の高いモデルを選ぶことが推奨されます。実際、最新のEmbeddingモデルの中にはOpenAIのモデルを凌駕する高精度を示すものも登場しています [10]。一方で、　**前処理やチャンク化の最適化だけで検索精度の7〜8割が決まる**　との指摘もあり [11]、モデル選択とあわせてデータ準備面のチューニングが重要です。

チャンク化戦略

- **文字数（トークン数）チャンク**: ドキュメントを一定の文字数・トークン数ごとに区切るシンプルなチャンク戦略です。例えば300〜500トークンを目安に固定長で切り出す方法で、実装が容易かつ個々のEmbeddingベクトルの長さが安定するメリットがあります [12]。その反面、章や文の途中で機械的に分割されることで**意味が途切れてしまう懸念**があり [12]、内容によっては回答時に文脈が失われるリスクがあります。
- **セマンティックチャンク**: ドキュメントの内容構造に沿って、**意味が壊れないようにチャンク化**する戦略です。例えばチャンク間で**重複部分（オーバーラップ）**を持たせることで境界の情報損失を減らし、文脈の連続性を保つ方法があります [13]（例: 一つのチャンクを350トークンとし50トークンを次チャンクに重ねる）。あるいはMarkdownや見出し階層に基づいて**論理的な単位で区切る構造化チャ**

ンクも有効です [14] 。章・節ごとにまとまりを保ちつつ、長さが長すぎる場合は再度300〜600トークン程度に小分けします [14] 。セマンティックチャンクは実装の手間や若干のコスト増を伴いますが、**論理一貫性が高まり検索ヒット率や回答品質の向上**に顕著な効果があります [15] 。

- **他にはどんなものがある？**: チャンク化については他にも様々なアプローチが模索されています。例えば、**動的なチャンク選択**では検索時には細粒度のチャンクでピンポイントに該当箇所を探しつつ、回答時にはそのチャンクの**親ドキュメント全体をコンテキストに利用する**手法があります [16] （上位n件中m件以上が同一文書に属する場合、その文書全体を参照するなど）。また、各ドキュメントの**要約（サマリー）を事前にEmbedding**して索引とし、まずサマリーレベルで候補絞り込みを行ってから詳細チャンク検索する二段階の検索も行われています [17] 。これらにより、広い文脈を保持しつつ効率よく関連情報を取得する工夫がなされています。

Retrievalフェーズ

検索戦略

- **ベクトル検索**: ユーザの質問やドキュメントをベクトル（埋め込みベクトル）に変換し、**高次元空間における類似度**にもとづいて関連文書を検索する方法です。各ドキュメントをあらかじめベクトル化してデータベースに格納し、質問文もベクトル化して**近いベクトルを持つ文書**を近似最近傍探索(ANN)で高速に見つけ出します [18] 。ベクトル検索は単語の表面上の一致ではなく**意味的な近さ**にもとづくため、ユーザ問いに対し直接的なキーワードが含まれない関連文書も発見できる利点があります。
- **フルテキスト検索**: 単語やフレーズの**文字列一致**にもとづいて文書を検索する従来型の方法です。典型的にはTF-IDFやBM25といったスコアリングアルゴリズムを用いたキーワード検索が該当します [19] 。ベクトル検索に比べシンプルですが、誤字脱字に弱い一方で**特定キーワードの厳密な含有**を絞り込む用途には適しています。実運用ではベクトル検索で漏れがちな固有名詞マッチや数値データの検索に併用されることがあります。
- **セマンティック検索**: クエリとドキュメントの**意味的関連性**にもとづいて結果を返す検索手法の総称です。ベクトル検索もセマンティック検索の一種ですが、ここでは**より高度な文脈理解**を用いる検索を指します。例えばAzure Cognitive Searchのセマンティック検索では、クエリと文書内容をTransformerモデルで精査し**質問に対する直接的な答えになりうる文脈**を持つ文章を上位に返します [20] 。セマンティック検索は一般にキーワード検索より良い結果をもたらすと期待されていますが、対象や質問次第ではキーワード検索の方が適する場合もあり、両者を組み合わせた**ハイブリッド検索**が有効です [21] 。
- **メタデータフィルター**: ドキュメントに付与されたメタデータ（属性情報）を使って**検索対象を事前に絞り込む**手法です [22] 。例えば質問の意図に応じて「年度=2023」や「ドキュメント種別=技術仕様書」など条件を課すことで、最初から関係の薄いデータを除外できます。メタデータフィルタリングによって**ノイズを減らし検索精度を高める**ことができ、特に社内データ検索などでは有効な戦略です [22] 。
- **他にはどんなものがある？**: 上記の要素を**組み合わせた検索**も一般的です。例えば**ハイブリッド検索**ではベクトル類似度とキーワードマッチのスコアを統合して総合順位を決定します [21] 。また、最近では**RAG-Fusion**のようにLLMを用いて元の質問から**類似だが多様な複数クエリを自動生成し、それぞれ検索した結果をRRFで統合する**高度なアプローチも登場しています [23] 。このように検索段階での工夫により、ユーザの意図を幅広く汲み取って漏れの少ない関連情報を取得できるようになります。

リランク戦略

- **RRF（Reciprocal Rank Fusion）**: 複数の検索結果リストを**順位スコア融合**する手法です。一つのクエリに対し異なる検索器（例: ベクトル検索とBM25）が返した結果について、それぞれの順位に基づき定式化されたスコアを合算して**単一の統一ランキング**を作成します [24] 。RRFを用いることで、キーワード検索とベクトル検索の**双方の強みを活かした結果統合**が可能になり、検索の網羅性・精度を向上させることができます。

- **セマンティックランカー**: 検索後の上位文書を、**より精密なモデルで再評価（リランキング）**する手法です。クエリと各候補文書を対にして入力し関連度スコアを算出する**機械学習モデル**（クロスエンコーダ型のBERTモデルやGPT系モデルなど）を用いて、初期検索結果の順位を付け直します。例えばある事例では、**ハイブリッド検索で得た50件の候補をさらにLLMでセマンティック評価して上位順を改善**する2段階のアプローチが取られています [25]。セマンティックランカーにより質問意図に即した文脈の文書が上位に来るようになり、最終的な回答精度向上につながります。
- **他にはどんなものがある？**: リランキング手法は他にも様々な種類があります。例えば**Cohere社のRerank API**のようにあらかじめ学習済みの再ランクモデルをサービスとして利用できるものもあります [20]。また、LangChainの**Ensemble Retriever**やLlamaIndexの**Node Postprocessor**では、**LLMそのものに候補文書を評価させて並べ替える**ことも可能です [26]。用途に応じてこれら手法を組み合わせることで、必要な情報がより上位に現れるよう工夫できます。

Augmented Generationフェーズ

プロンプトエンジニアリング戦略

- **チャンクの最終選定**: 検索で取得した複数のチャンクから、実際にプロンプトに含めるコンテキストを選別する段階です。ここでは**質問に直接関連し信頼できる情報のみを含める**ことが重要です。具体的には、検索上位の1〜5件程度に質問に関連した正確な情報が含まれている状態が望ましく、複数チャンクを与える場合は**内容が相反しないようにする**べきだとされています [27]。これにより、LLMが矛盾なく回答を生成できるようになります。
- **メタデータの利用**: プロンプト内でコンテキストとして提示する際に、**メタデータ情報を活用する**ことも効果的です。例えばドキュメントの発行日や出典名をテキストに含めて提示すれば、モデルは「その情報がいつのものか」「どの資料から来ているか」を把握でき、**情報の新しさや信頼性を判断する手がかり**になります [28]。実際、ニュース記事の要約生成などでは日付メタデータを与えることで最新情報を優先して前面に出す効果が確認されています [28]。またユーザからの質問日時や属性に応じて、あらかじめメタデータフィルタで該当する情報だけを検索・コンテキスト化する運用も考えられます [29]。
- **他にはどんなものがある？**: この他、プロンプトの工夫として**LLMへの指示の明確化**が挙げられます。例えば「以下のコンテキストを使って質問に答えてください。答えがない場合は『知らない』と答えてください」といった指示をプロンプトに組み込むことで [30]、与えた情報に基づかない勝手な創作（ハルシネーション）を防ぎやすくなります。プロンプトエンジニアリングは試行錯誤のプロセスであり、ユーザからのフィードバックを踏まえて**プロンプトテンプレートを継続的に改善**していくことが、最終的な回答品質を高めるポイントです。 [31]

---

[1] RAGにおけるデータクレンジングの重要性｜Data Science Career Note
https://digital-transformation-blog.com/data-cleansing-rag/

[2] うさぎでもわかるRAGの精度を劇的に向上させる実践ガイド | 法人向け生成AIチャットのナレフルチャット
https://www.knowleful.ai/plus/rabbit-rag-accuracy-guide/

[3] [4] [5] [6] [11] [12] [13] [14] [15] RAG検索は前処理×チャンクで決まる｜ベクトル埋め込み精度を8割伸ばす実践ガイド
https://www.openbridge.jp/column/rag-preprocess-chunking

[7] [8] RAGにおける前処理の重要性と実践事例 - fenfenkunの日記
https://fenfenkun.hatenablog.com/entry/2024/10/25/070000

[9] RAGの性能を向上させるEmbedding Modelの選択 | ネットワンシステムズ
https://www.netone.co.jp/media/detail/20240823-01/

10 RAGエンベディングガイド｜最新モデル比較・コスト・精度〖2025年版〗
https://arpable.com/artificial-intelligence/rag-embedding-technology/

16 17 18 19 20 26 30 RAG (検索拡張生成) の応用手法 (パターン集) | @Subaru
https://netweblog.wordpress.com/2023/11/08/llm-advanced-rag-methods/

21 25 27 RAGを使ってLLMでも最新情報や企業内情報にも対応する | ネットワンシステムズ
https://www.netone.co.jp/media/detail/20231006-01/

22 29 生成AI(RAG)の利活用、うまくいってますか？RAG精度向上のポイントを4点まとめてみた。｜テクニカルブログ｜日本情報通信株式会社
https://www.niandc.co.jp/tech/20241220_57019/

23 24 RAG Fusionが思ってたより凄そう
https://zenn.dev/ozro/articles/abfdadd0bfdd7a

28 31 RAG の最適化: より優れたデータとプロンプトで LLM を強化する | Shaip
https://ja.shaip.com/blog/rag-optimization-with-data-and-prompts/