RAG として AIChatBot に与えるデータベースの構造に関する一考察

On Database Structures for Retrieval-Augmented Generation in AI Chatbots

神奈川工科大学
川越 航太

概要 (Overview)

- RAG とベクトル DB の概要
- データセット & ファイル形式
- 比較実験: 設定と結果
- 考察・結論と今後の課題

研究背景と目的

1 目的（Goal）

- RAGベースのAIチャットボットを構築し、顧客の質問に自動回答してサポート業務を効率化する。
- 応答ログを活用した、新人オペレーター等の社内教育への活用。

2 背景（Background）

- オペレーターと顧客のやり取り、ユーザマニュアルなどから自動生成したFAQを、LLMが正確にリトリーブするためにどうすればよいか。

3 データ提供元

- 日本のEdTech企業で職能団体・学会・自治体などの研修をオンライン／オフライン問わず一元管理できる「研修DXプラットフォーム」を提供。

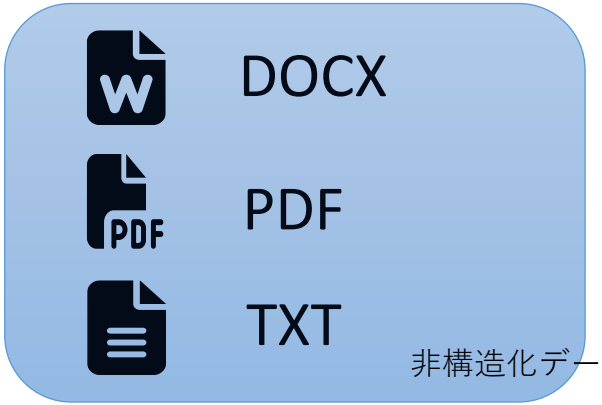
RAGシステムのフロー

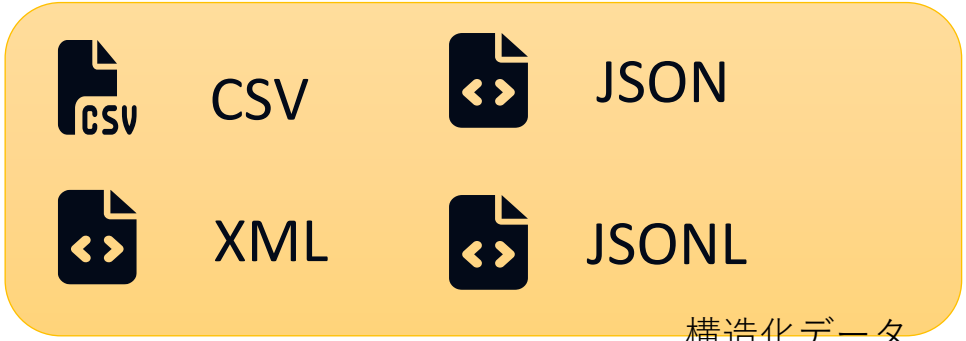
大規模言語モデルは外部知識ベースから関連情報を検索し、回答生成に活用する。

RAG (Retrieval-Augmented Generation) は、クエリに対してベクトルデータベースから類似文書を検索し、LLM へ渡して応答を生成する仕組み。

本研究では FAQ データセット (100件/1000件) を複数形式に変換し、VDB 化して検索精度を評価した。

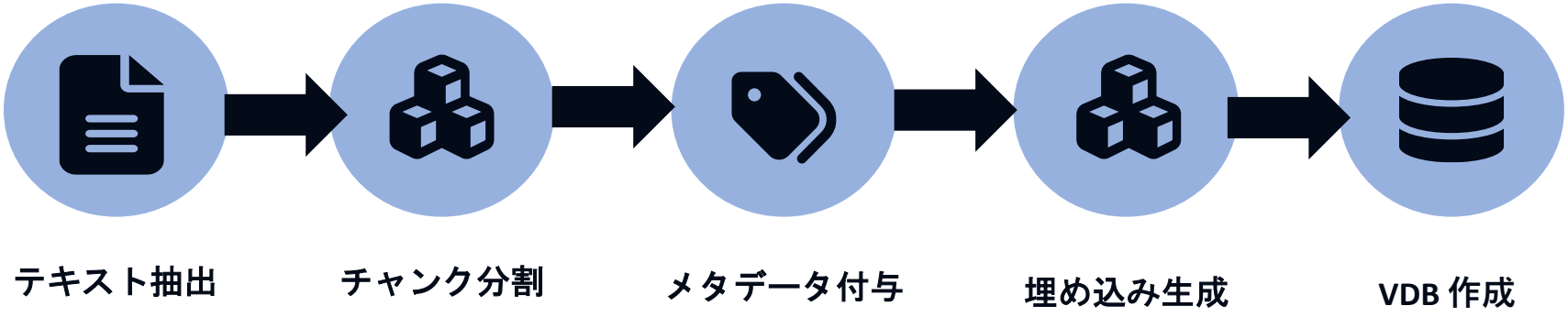
データセットとファイル形式





形式	抽出ライブラリ	分割単位	メタデータ
DOCX	python-docx	段落	ファイル名+段落番号
PDF	PyPDF2	ページ	ファイル名+ページ
TXT	open().read	全文	ファイル名
CSV	pandas.read_csv	行	ファイル名+行番号
XML	xml.etree	エントリ要素	ファイル名+エントリ番号
JSON	json.load	オブジェクト	ファイル名+エントリ番号
JSONL	json.loads	行	ファイル名+行番号

前処理パイプライン



- チャンク長: 1000 文字, オーバーラップ: 200 文字
- 埋め込みモデル: intfloat/multilingual-e5-large
- 使用ライブラリ: LlamaIndex
- 言語モデル: Gemini 2.0-flash

非構造化データのVDB化の例

チャンク

オーバーラップ

Question:無料トライアル期間が終了した後、自動的に有料プランへ移行され、課金が開始されますか？
Answer:いいえ、自動で有料プランへ移行されることはないです。トライアル期間終了後に引き続きご利用いただく時は、管理者設定画面から明示的にクレジットカード情報を登録し、プランのアップグレード手続きを行っていただく必要があります。
Question:経費精算の申請が承認者から差し戻された場合、元のデータを修正してそのまま再提出することはできますか？
Answer:はい、可能です。「申請一覧」の「差し戻し・却下」タブから該当の申請データを開き、指摘された箇所（金額や領収書の添付など）を修正した上で、再度「申請する」ボタンを押下してください。
Question:受講者がログインパスワードを忘れてしまったら、管理者がシステム側から代わりに再設定することは可能ですか？
Answer:はい、可能です。管理者画面の「ユーザー管理」から対象の受講者を選択し、「パスワードのリセット」を実行してください。受講者の登録メールアドレス宛に、パスワード再設定用の専用URLが自動送信されます。

構造化データのVDB化の例

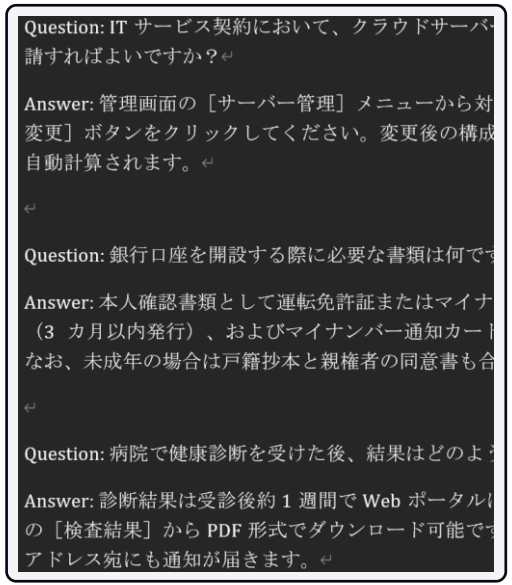
質問

回答

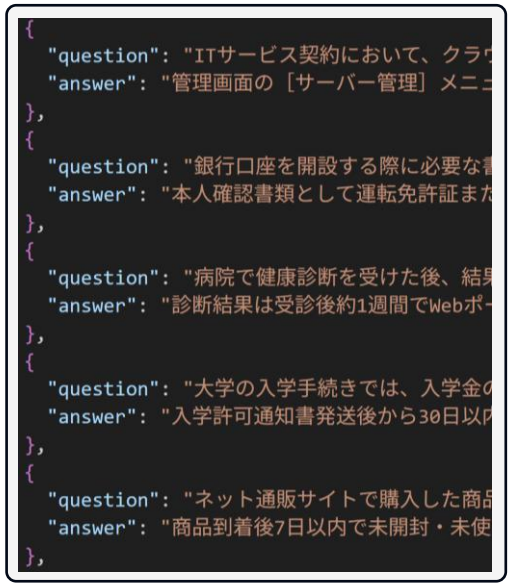

```
[
 {
  "question": "無料トライアル期間が終了した後、自動的に有料プランへ移行され、課金が開始されますか？",
  “answer”: “いいえ、自動で有料プランへ移行されることはないです。トライアル期間終了後に引き続きご利用いただく時は、管理者設定画面から明示的にクレジットカード情報を登録し、プランのアップグレード手続きを行っていただく必要があります。"
 },
 {
  “question”: “経費精算の申請が承認者から差し戻されたら、元のデータを修正してそのまま再提出することはできますか？ ",
  "answer": "はい、可能です。「申請一覧」の「差し戻し・却下」タブから該当の申請データを開き、指摘された箇所（金額や領収書の添付など）を修正した上で、再度「申請する」ボタンを押下してください。"
 },
```

ファイルの内容と構造例

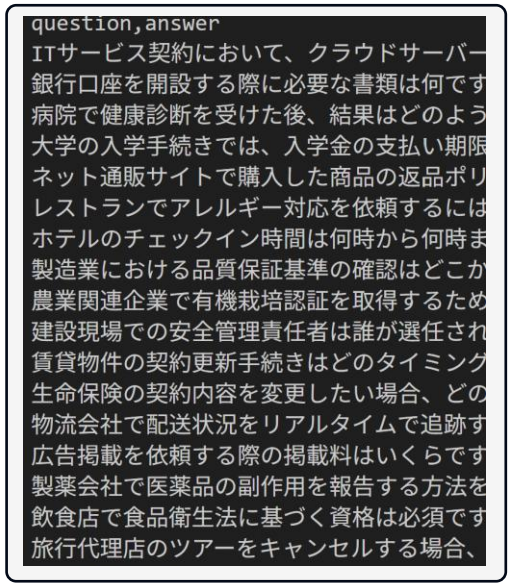
DOCX



JSON



CSV

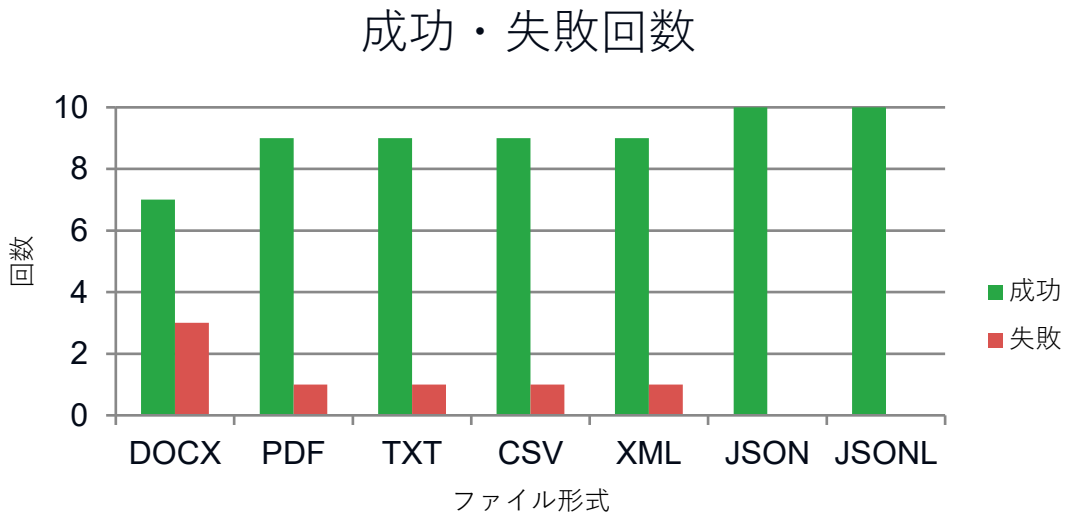


各形式に応じたチャンク構造:
DOCX は段落単位、CSV は行単位、JSON はオブジェクト単位に分割される。
100問はEdTech企業からの提供データ、+900問はそれを基にOpenAI o3で拡張したデータ。

実験設定① 100 問に対する10問の質問

1	管理者側から受講者を申し込んだ場合、受講者には申込み完了のメールが送信されますか？
2	クレジットカード払いでキャンセルした場合、返金はいつ頃になりますか？
3	代表者アカウントのメンバー画面で「メンバー情報変更」と「アカウントの削除」を制限するにはどうすれば良いですか？
4	修了証はどのようにダウンロードできますか？
5	コースの言語設定を変更できますか？
6	受講者の進捗状況を一括で更新できますか？
7	システムメンテナンス情報はどこで確認できますか？
8	講師の都合で講義の日程や順序を変更する方法を教えてください。
9	受講料の一部を振り込んだ後、残りの金額を支払う方法を教えてください。
10	振込コードを入力しなかった場合のリスクは何ですか？

100 問実験: 成功 / 失敗回数



- JSON / JSONL では、全問正解 (10/10)。
- DOCX は 70% (7/10) と低調。
- その他の形式は 90% (9/10) の精度を示した。

考察 (100問)

• DOCX 形式は段落単位の分割が適切に働かず、レイアウトやネスト構造の影響で正答率が低下した。

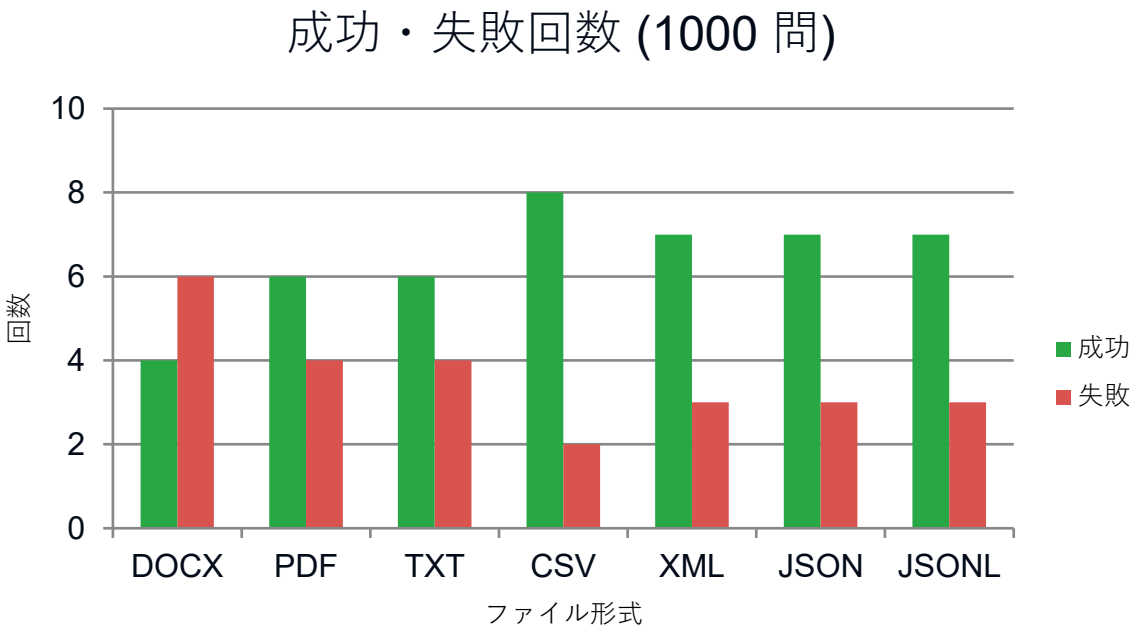
• JSON / JSONL はオブジェクト/行単位に分割されたので、構造化度の高さが有利に働いた。

• PDF・TXT・CSV・XML は 90% と高精度。シンプルな構造が検索性能を上げた。

実験設定② 1000 問に対する10問の質問

1	レッスンに紐づく課題の回答ボタンが表示されません。どの受講状況で回答可能になりますか？
2	施設側から除名すると、管理画面上ではどうなりますか？
3	ログインセッションのタイムアウト時間を変更できますか？
4	オーダーIDのない支払いカードが生成されることがありますが、どういう場合に起こりますか？
5	二段階認証を有効化する方法は？
6	アカウントを削除（退会）したい場合は？
7	ユーザーのログイン履歴を確認できますか？
8	アップロードファイルのサイズ制限は？
9	イベント発生時のWebhook通知は設定できますか？
10	データ保持ポリシー（GDPR対応）の設定方法は？

1000 問実験: 成功 / 失敗回数



・CSV が 8/10 で最高成績を維持。

・一方、JSON 系は 7/10、DOCX は 4/10 と大きく低下した。

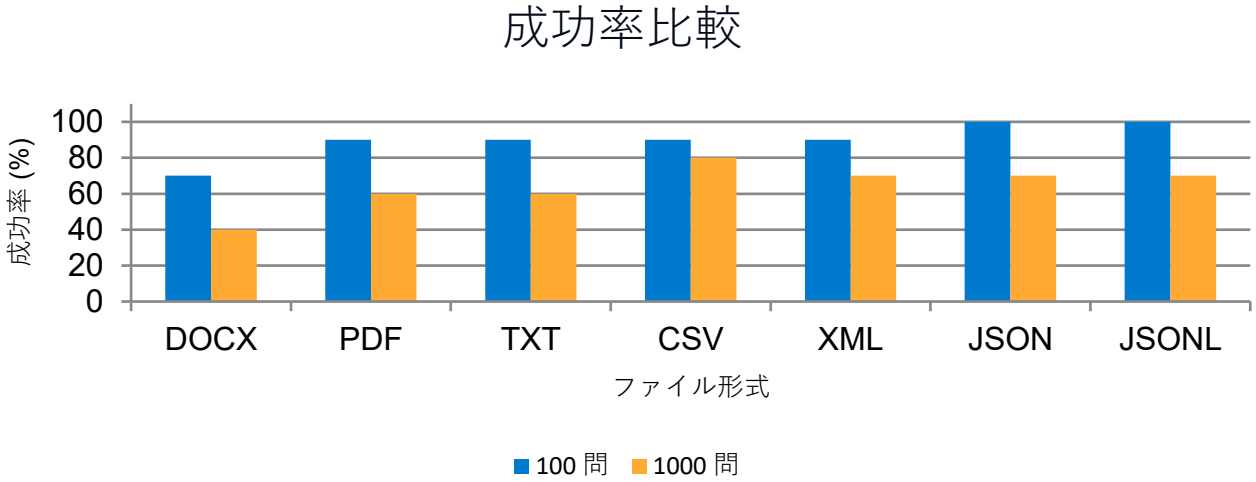
考察 (1000問)

- データの増加に伴い、検索成功率が低下。

- 多くなったデータを３つのファイルに分割して保存したが、100問での実験の３倍以上の問題数を１つのファイルにまとめた。100問ずつに分割すれば、検索精度は高まった可能性が残る。

- ファイルのカテゴリでの分割手法を用いなくても、CSV,JSONなどの構造化データは一定の精度を維持できたので、構造化ファイルはQAシステムのRAGにとって相性が良い。

実験比較



- JSON 系は 100 問では 100% だったが 1000 問では 70% に低下。
- CSV は安定して 80% を維持し、最も堅牢であることが示された
- DOCX は大規模化で 70%→40% と急落し、レイアウト依存性の脆弱さが顕在化した。

形式別の検討

DOCX

レイアウトや段組み、構造化データではないので、チャンク化が難しく、大規模時に成功率が低下。

JSON/JSONL

オブジェクト単位に分割でき、データ構造がしっかりと固定されていてVDB化しやすい。

TXT

全文を単一の文章として扱うため文脈が保持されるが、メタデータが少なくバリエーションに弱い。

CSV

レコード単位での分割が自然で、構造が単純なため大規模でも高精度を維持。

PDF

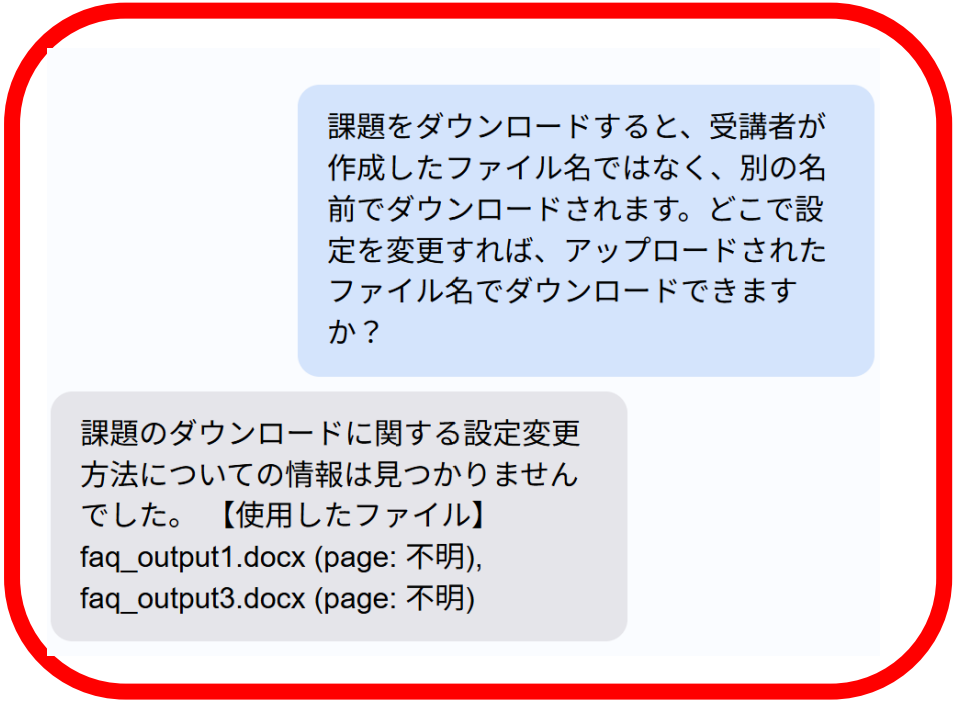
PyPDF2 による改行検出が不十分で、空白・タブの揺れが検索を阻害する可能性がある。

XML

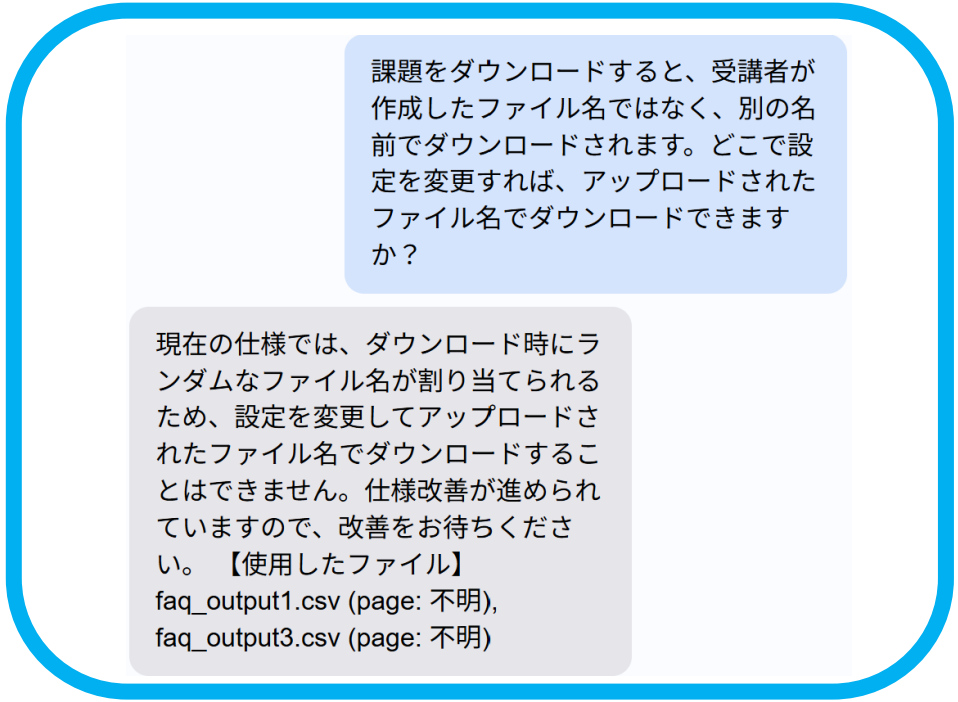
オブジェクト単位でデータを分割して管理できる。統一されたデータ構造により、安定した解析や変換が可能。

ファイル形式の違いによる失敗例と成功例

ハルシネーションについて：1000問程度では誤った回答を生成することはなく、検索が失敗した場合には、「分からない」と出力した。



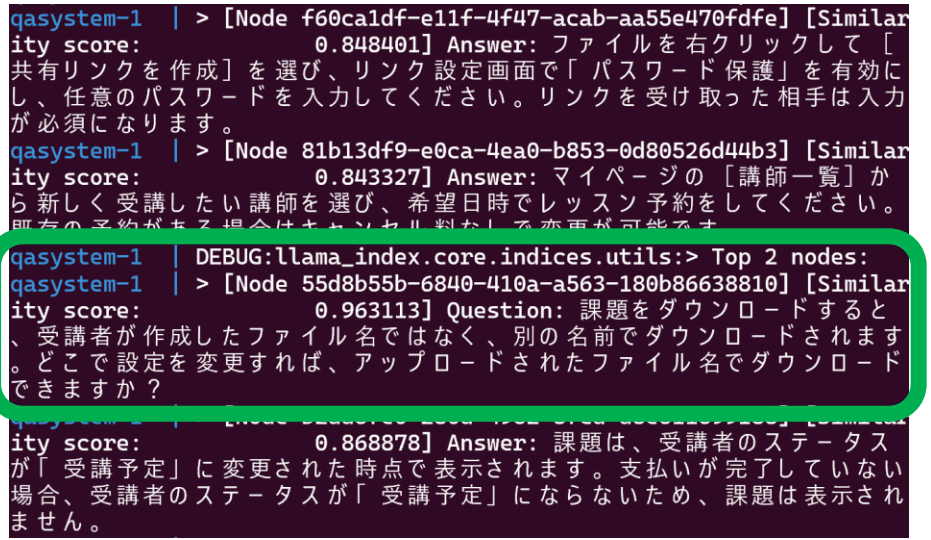
DOCXでの失敗例



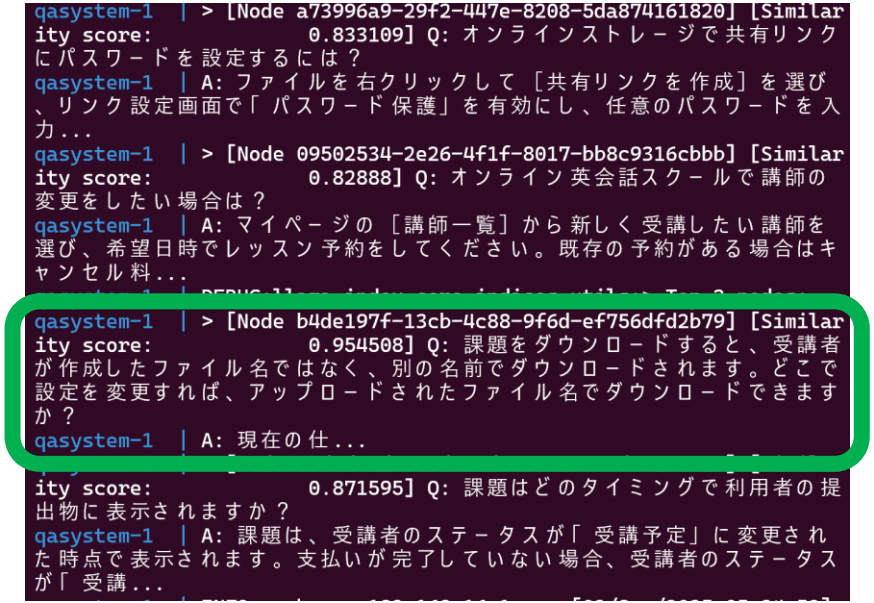
CSVでの成功例

検索精度に差が生じた原因

質問部分は両方とも検索に成功しているが、DOCX形式では質問と回答の紐付けが出来ないケースが散見され、これが原因となって回答を探すことに失敗している。

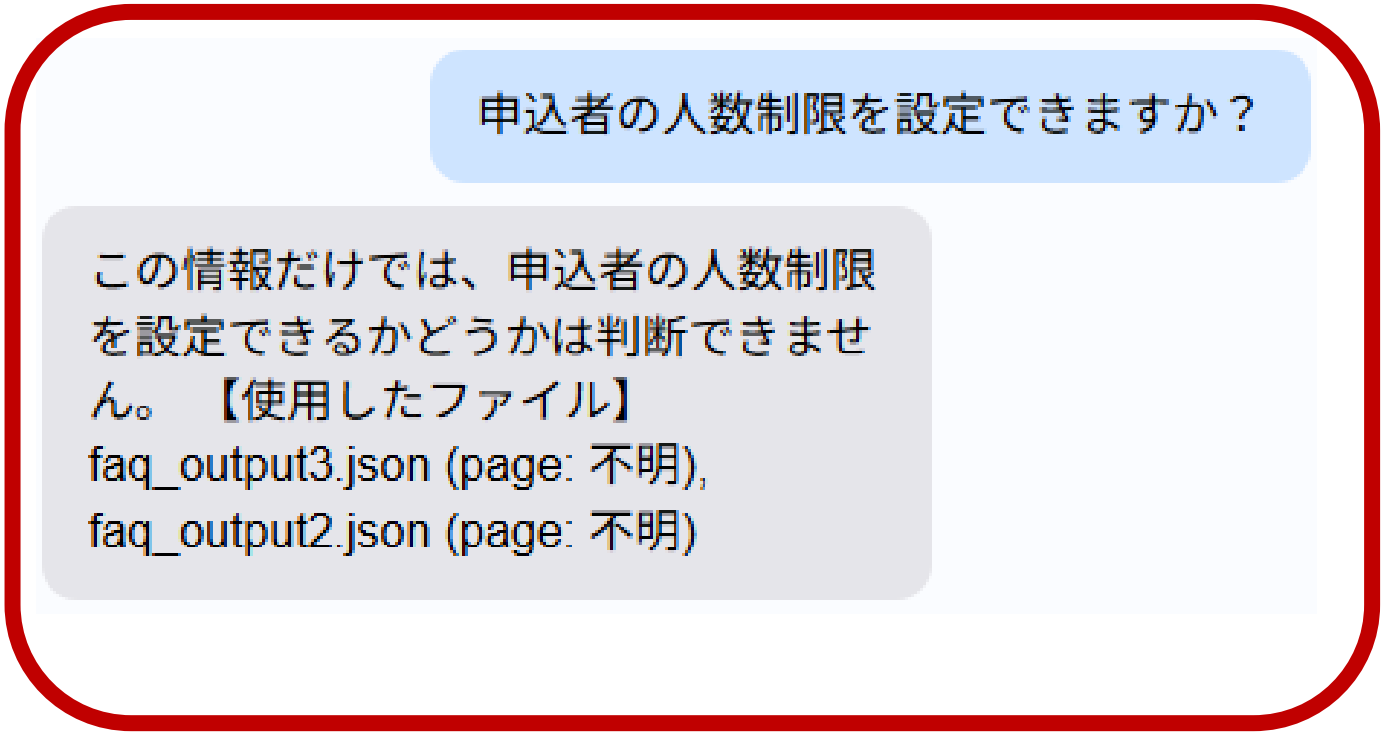


DOCXでのログ



CSVでのログ

その他の失敗例

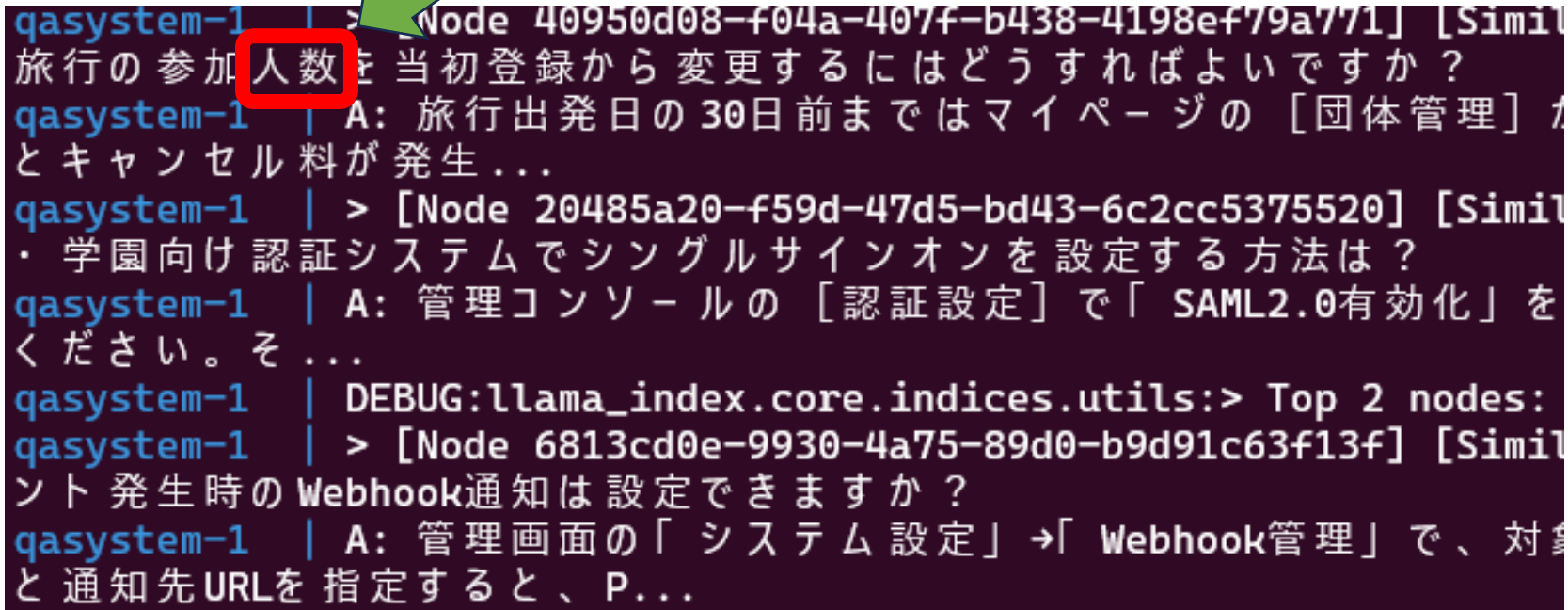

申込者の人数制限を設定できますか？
この情報だけでは、申込者の人数制限を設定できるかどうかは判断できません。 【使用したファイル】
faq_output3.json (page: 不明),
faq_output2.json (page: 不明)

失敗例

その他の原因

失敗のパターン：クエリの内容に近い単語が類似度が高くなりピックアップされるが、関係のないデータである場合が多い。

申込者の人数制限を設定できますか？



運用ベストプラクティス & 結論

- DOCX, PDFなどから構造化形式（CSV、JOSN等）へ統一
- VDB化する前に、元ファイルのクレンジング＆スキーマ自動検出
- 構造化データにできない場合には、適切なチャンク長/オーバーラップを設計
- ファイル名などのメタデータの付与で検索文脈保持
- 単純なファイル分割ではなく、QAのカテゴリごとにファイルやディレクトリを分割することで、誤検出を減らす
- 大規模での運用は「自動前処理 +構造化」で実装

限界と今後の課題

- データセット規模が限定的 (1000 FAQ) であり、一般化にはさらなる検証が必要。
- 埋め込みモデルや LLM の種類を変えた場合の評価は未実施。
- 固定チャンク長のため動的なチャンクサイズ調整の効果を検証できていない。
- 今後はクレンジング・スキーマ検出の自動チューニングやエラー検出機構の導入を検討。

Thank you!

ご清聴ありがとうございました

Questions?