文書画像 QA の表理解における検索と生成に適した表現形式の提案

蒲原悠登 [1]　竹内孔一 [2]
[1] 岡山大学大学院 環境生命自然科学研究科
[2] 岡山大学大学院 環境生命自然科学学域
p3z19g5j@s.okayama-u.ac.jp　takeuc-k@okayama-u.ac.jp

概要

文書画像 QA において、表は構造と内容の理解が必要であるため難易度が高い。本研究では、表のテキスト表現（Markdown、要約、併用）が MLLM-RAG の検索と生成に与える影響を検証し、検索と生成に適した表現形式の提案を行う。JDocQA を用いた実験の結果、検索では構造と意味を補完する「併用」が最も有効であった。一方、回答生成では、参照外画像のノイズに対し、画像に加え Markdown をテキスト提示することが頑健性維持に有効であることを示した。これらに基づき、検索と生成の各段階に適した表テキスト表現の設計指針を提案する。

1　はじめに

文書画像に対する質問応答（Document VQA）は，帳票・マニュアル・パンフレット等の実務文書を対象に重要性が高まっている．[1] 中でも表は，行列構造に基づく数値・属性の対応関係を含み，質問が参照すべきセルや列を正しく特定できなければ誤答に直結する．さらに，文書画像はテキストだけでなく，ページ構造や非テキスト要素（図・区切り等）を含む多様な要素が混在するため、表への質問は「どの表を見るべきか」と「表から何を読み取るべきか」の両方ともが難しい課題である．

近年，検索と生成を組み合わせた RAG は知識集約タスクで有効な枠組みとして整理されており，[2] これをレイアウトや図表を含むマルチモーダル文書へ拡張した MLLM-RAG も提案され，[3] 注目されている．ただし，この枠組みが有効に機能するためには，検索段階で適切な根拠を供給できること，および生成段階でその根拠を正しく参照できることの両方が満たされる必要がある．表への質問の失敗要因は二段階に分かれる．第一に，検索段階で正しい表を上位に提示できなければ，生成はそもそも根拠不足となる．第二に，正しい表を提示できても，生成段階で表の構造理解・数値の取り違え・注意の分散が起きると誤答が生じる．特に，関係のない画像が混入しうる現実的設定では，画像のみの入力は頑健性が低下しやすい一方で，表をテキスト化して，テキスト表現として併用することで注意を誘導できる可能性がある．

本研究では，表を Markdown（MD）として構造表現に変換する方法，表内容を要約（SUM）として言語化する方法，および MD+SUM の併用が，検索・回答の双方に与える影響を体系的に評価する．具体的には，レイアウトに基づくチャンク化と埋め込み検索により、どのテキスト形式の表がヒットしやすいかを測定し，さらにマルチモーダル生成モデルに対しての画像の入力枚数など、入力内容を変え，回答精度の変化を分析する．

本論文の貢献は以下の 2 点である．

- 表のテキスト表現（MD/SUM/MD+SUM）を，検索（Hit@5）と回答（LLM-as-a-Judge の正解率）の両面から比較し，検索と生成という目的に応じた設計指針を示す．
- 総画像枚数（3/5/10）と表テキスト表現を要因とする実験により，画像のみ入力の劣化とテキスト併用の頑健性を定量的に明らかにする．

2　関連研究

文書に対する質問応答は，帳票・マニュアル・パンフレット等の実世界文書を対象とし，視覚情報とテキスト情報の両方を統合して解く必要があるため，近年注目を集めている．代表的なデータセットである DocVQA は，既存 VQA/読解モデルに対して依然として大きな性能ギャップが残ることを示し，とりわけ文書構造の理解が重要な質問で性能が落ち

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

る点を報告している．[4]

また視覚情報を直接解釈して推論することは難しく，画像を構造化テキストへ変換してから推論する枠組みが有効であることが報告されている．DePlot は，チャート画像を線形化した表へ変換し，その出力に対して LLM で推論する 2 段階（変換→推論）に分解することで，高い性能を得られることを示した．[5] この構造化テキストへ落として推論するという考え方は，表への質問に対しても自然に適用できると考えられる．すなわち，表の画像を Markdown 等の構造を保持したテキスト表現に変換すれば，生成モデルが表の行列関係を参照しやすくなり，また検索段階でも表の内容を埋め込み空間で比較しやすくなる可能性がある．本研究はこの観点から，Markdown（MD），要約（SUM），および両者を併用した MD+SUM として表現し，これらを検索・回答の両段階で利用する表テキスト表現設計を提案する．その上で，同一の設定で各表現の影響を比較し，ノイズ混入や入力枚数の変化といった現実的条件下での頑健性を検証する．

3 手法

本研究では，表画像に対する質問応答における MLLM-RAG のために，表チャンクを構造と意味の観点からテキスト化して活用する表現設計を提案する．具体的には，表領域をレイアウト解析で抽出した上で，(i) 行列構造を保持するのを目的とした、表を変換した Markdown（MD），(ii) 表の意味・要点を自然言語で補完するのを目的とした要約（SUM），(iii) 両者を連結した MD+SUM，の 3 種の表テキスト表現を構成し，検索および生成入力に利用する．

また、失敗要因を切り分けて分析できるよう，MLLM-RAG の性能変化を (i) 必要な表チャンクを上位に取得できるかという検索面，(ii) 取得した根拠を参照して正しく回答できるかという回答面，の 2 段階に分解して評価する．

3.1 全体パイプライン

まず PDF をページ画像に変換し，文書解析ライブラリの Yomitoku[1)] を使って OCR とレイアウト検出を行う．検出された bbox 領域を 1 チャンクとし，ページ番号・bbox 座標などのメタ情報とともに保存する．検索実験では，チャンク集合を DB として質問文から埋め込みを使った検索を行い，Top-5 に質問の回答に必要な表チャンクが含まれるかを測定する．回答実験では，質問が参照したページの画像と参照外ページ画像を並べ，加えて正解 OCR テキストと不正解 OCR テキストを併記した入力を MLLM へ与え，ノイズがある中での回答精度を評価する．

3.2 チャンク化と表テキスト表現

Yomitoku のレイアウト検出結果に基づき，ページ内の各 bbox を 1 チャンクとする．各チャンクは PDF 識別子，ページ番号，bbox 座標，OCR テキストを持つ．表チャンクについては，以下の 3 種類のテキスト表現を用意する．(i) **MD**: 表構造を Markdown 表に変換したもの，(ii) **SUM**: 表内容を自然言語で要約したもの，(iii) **MD+SUM**: 両者を連結したものである．MD は構造保持，SUM は表の意味が説明できるという理由から有用である可能性があり，MD+SUM は両者のメリットをどちらも活かすことを意図する．また要約は `Llama-3-ELYZA-JP-8B`[6] を使い、表をマークダウンにしたものとその表の存在したページ内のテキストをモデルに入力し、出力させることで作成した．

3.3 回答に必要な表チャンクの同定

検索評価には各質問が依拠する正解チャンクが必要である．JDocQA は参照 PDF ファイルと参照ページを提供する一方，ページ内でどの部分が根拠かは明示していない．そこで本研究では参照ページ内の表チャンクを候補とし，(1) 表チャンクが 1 つならそれを正解チャンクとし，(2) 複数ある場合は各候補表（MD 等）を入力として ELYZA-JP-8B に「質問に答える根拠として最も適切な表」を選択させ，選ばれた表チャンクを正解チャンクとする．この手順により，検索結果に正解チャンクが含まれるかで検索精度を評価することができる．

4 実験設定

4.1 データ

JDocQA の表への質問 337 件を対象とし，参照 PDF ファイルと参照ページが付与された QA 集合から評価用データを構成する．各 PDF はあらかじめ Yomitoku を使った OCR・レイアウト解析を行い，ページ内 bbox 単位のチャンク集合として保存し，検索 DB および生成入力の素材として用いる．

1） https://github.com/kotaro-kinoshita/yomitoku

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

4.2 検索実験

検索実験では，質問文から関連チャンクをベクトル検索で取得し，Top-5 に正解チャンクが含まれるかで評価する．チャンクの埋め込みには `cl-nagoya/ruri-v3-70m`[7] を用い，質問文および各チャンクテキストをベクトルに変換して近傍検索を行う．検索対象は OCR・レイアウト解析で得た全チャンクであり，表チャンクについては 3.2 節の表テキスト表現をチャンクテキストとして用いる．比較条件は，表テキスト表現の MD/SUM/MD+SUM の 3 種類である．

4.3 回答実験

回答実験では，MLLM として Qwen2-VL-7B[8] を用い，参照ページ画像と参照外のページ画像群とテキストを並べた入力から回答を生成させ、その回答を LLM-as-a-Judge で評価する．また参照ページ画像は常に入力の先頭に配置しており、画像順に対応するようテキストを並べて配置している．比較する条件は 3 つあり，(i) 参照画像の位置（先頭/中央/末尾），(ii) 総画像枚数（3/5/10），(iii) テキスト種類（画像のみ／MD ／ SUM ／ MD+SUM）を組み合わせて比較する．テキスト種類が「画像のみ」の条件では生成入力に表テキストを付与しない．それ以外の条件では，正解表チャンクに対応するテキスト表現を入力へ付与する．

4.4 評価指標

検索実験では Hit@5 で評価する．回答実験では，参照解答と生成解答の整合性を GPT-4o による LLM-as-a-Judge で判定し，正解率（Accuracy）を指標とする．

5 結果と考察

5.1 検索実験の結果

表 1 に，表チャンクのテキスト表現を切り替えた場合の検索性能（Hit@5）を示す．

5.1.1 検索実験結果に対する考察

MD は表の行列構造を保持する一方で，質問文と語彙が一致しない場合があり，検索では必ずしも最良にならない．SUM は自然言語での言い換えにより質問との使用している単語の類似度を高めるが，構造情報が失われる．これに対し MD+SUM は，単語の類似度（SUM）と構造保持（MD）を同時に与えるため，最も高い性能を示した．この結果は，表質問に対する検索では，質問文と対応づけやすい自然言語的手がかりが重要であり，ただし構造情報も補助として有効であることを示唆する．

表 1 検索性能（Hit@5）．Top-5 に正解チャンクが含まれる場合を成功とする．

表テキスト表現	Hit@5
MD	0.347
SUM	0.436
MD+SUM	0.501

表 2 正解を初めに置いた場合の回答精度（LLM-as-a–Judge 正解率）．

テキスト種類	3 枚	5 枚	10 枚
画像のみ	0.519	0.501	0.504
MD	0.519	0.501	0.522
SUM	0.492	0.507	0.486
MD+SUM	0.540	0.522	0.516

5.2 回答実験の結果

表 2-4 に，回答精度（LLM-as-a-Judge 正解率）を示す．画像のみ条件は，総画像枚数が増えるほど正解率の低下が大きく，参照外ページ画像による注意の分散の影響を受けやすい．これに対し，MD，MD+SUM のいずれかを併用する条件では，画像枚数の増加に対する劣化が緩やかであり，マークダウンテキストにより情報を補完することができることが示唆される．

5.2.1 回答実験結果に対する考察

表 2-4 は，参照画像の提示位置と総画像枚数，および表テキスト表現が回答精度に与える影響を示す．全体として，(i) 正解ブロックを先頭に置いたときが最も高精度で，中央・末尾になるほど低下する，(ii) テキスト表現の中では MD および MD+SUM が比較的安定し，SUM は一貫して不利，という傾向が観察される．MD は中央/最後に置いた場合で特に改善が見られ，例えば最後に置いた場合では画像のみ 0.448（10 枚）に対し MD は 0.501 と大きく上回る．これは，Markdown が表の行列構造とセル内容を明示するため，モデルが視覚的な情報を取り切れなくても，言語側で根拠を再確認できるためと考えられる．MD+SUM も全体として安定して高いが，

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

表 3 正解を中央に置いた場合の回答精度（LLM-as-a–Judge 正解率）.

テキスト種類	3 枚	5 枚	10 枚
画像のみ	0.492	0.451	0.451
MD	0.504	0.519	0.492
SUM	0.471	0.462	0.397
MD+SUM	0.501	0.519	0.498

表 4 正解を最後に置いた場合の回答精度（LLM-as-a–Judge 正解率）.

テキスト種類	3 枚	5 枚	10 枚
画像のみ	0.480	0.454	0.448
MD	0.513	0.501	0.501
SUM	0.465	0.442	0.424
MD+SUM	0.504	0.480	0.492

先頭に置いた場合では 3 枚で 0.540 と最大値を示す一方，10 枚では 0.516 と MD（0.522）に僅かに劣る．これは，要約が追加されることで有用な語彙手掛かりが増える反面，入力が長文化し，画像枚数が 10 枚だと追加テキスト自体が注意分散を招く可能性がある．対照的に SUM は全条件で低く，特に 10 枚で中央に配置した場合では 0.397 まで低下する．要約は構造情報を圧縮するため，数値や列対応といった表質問の根拠に必要な情報が落ちやすい．また，要約文が誤って一般化したり，質問と部分的に語彙が合致してしまうと，もっともらしいが誤った根拠として作用し，誤答を誘発する可能性がある．

6 おわりに

本研究では，表質問に対する MLLM-RAG において，表のテキスト表現（MD/SUM/MD+SUM）が検索と回答生成の双方に与える影響を検証した．JDocQA の表質問 337 件を対象に，Yomitoku による OCR・レイアウト解析で bbox 単位のチャンク DB を構築し，検索は ruri-v3-70m によるベクトル検索，回答は Qwen2-VL-7B による生成を用いて評価した．

検索実験（Hit@5）では，MD（0.347）や SUM（0.436）に比べ，MD+SUM（0.501）が最も高い性能を示した．この結果は，検索段階では質問文と対応づけやすい自然言語的手がかり（SUM）が重要である一方，表の構造情報（MD）も補助として有効であり，両者の併用が有利であることを示唆する．

回答実験では，画像のみ入力は総画像枚数の増加に伴って性能低下が大きく，参照外ページ画像による注意分散の影響を受けやすいことが確認された．一方で MD や MD+SUM を併用する条件は，特に正解画像が中央・末尾に置かれる不利条件において改善が見られ，表の構造を保持したテキスト表現が根拠参照を補助し，ノイズ下での頑健性を高める可能性が示された．

以上を踏まえ，本研究から得られる知見を以下にまとめる．

- 検索段階では，表の要約を含む表現が有利であり，特に MD+SUM が最も高い検索精度となる．
- 回答生成段階では，画像のみ入力はノイズや入力長に脆弱であり，MD および MD+SUM を併用することで性能劣化を緩和できる．

本研究の限界として，(i) 正解表チャンク同定が LLM 選択に依存しており誤りが混入しうる点，(ii)SUM が要約の忠実性や粒度に左右される点，(iii) 表以外（図・本文主体の質問）への一般化が未検証である点が挙げられる．今後は，根拠領域のより厳密な同定，および学習型 retriever や画像埋め込み型 retrieval の導入による検索・生成の統合最適化を検討する．

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).

謝辞

議論に参加して下さいました竹内研究室の諸氏に心より感謝致します. 本研究は JSPS 科研費 JP22K00530 の助成を受けたものです.

参考文献

[1] Eri Onami, Shuhei Kurita, Taiki Miyanishi, and Taro Watanabe. JDocQA: Japanese document question answering dataset for generative language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 9503–9514, Torino, Italia, May 2024. ELRA and ICCL.

[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, 2024.

[3] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents, 2025.

[4] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021.

[5] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. Deplot: One-shot visual language reasoning by plot-to-table translation, 2023.

[6] Masato Hirakawa, Shintaro Horie, Tomoaki Nakamura, Daisuke Oba, Sam Passaglia, and Akira Sasaki. elyza/llama-3-elyza-jp-8b, 2024.

[7] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese General Text Embeddings, 2024.

[8] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. **arXiv preprint arXiv:2409.12191**, 2024.

This work is published without peer review and is licensed by the author(s) under CC BY 4.0 (https://creativecommons.org/licenses/by/4.0/).