テキストおよび画像を用いた検索拡張生成のための意味的文書レイアウト解析

植田 暢大 [1*] 董 于洋 [2*†] Krisztián Boros [1] 伊藤 大輝 [1] 世良 拓也 [1] 小山田 昌史 [1]
[1]NEC [2]SB Intuitions 株式会社
{nobuhiro-ueda,krisztian-boros,ito-daiki,takuya-sera,oyamada}@nec.com
yuyang.dong@sbintuitions.co.jp

概要

図表を含む文書ページの画像をトピック単位で分割する意味的文書レイアウト解析を提案する．検索拡張生成（RAG）で図表を含む文書を扱う際，ページ全体を視覚言語モデル（VLM）に入力する従来手法では情報過多から読解の誤りが起きやすい．意味的文書レイアウト解析を前処理に用いることで必要な文脈を保ったまま情報を削減でき，VLM によるより正確な読解が可能となる．本研究では，本タスクのためのデータセットを構築し，テキストおよび画像を用いた RAG ベンチマークにおいて手法の有効性を示した．

1 はじめに

検索拡張生成（Retrieval-Augmented Generation; RAG）は，大規模言語モデル（LLM）や視覚言語モデル（VLM）の回答の際に関連する外部知識を検索して使用する手法であり，回答の正確性を向上させる [1, 2, 3]．典型的な外部知識として，Web ページ，マニュアル，学術論文，企業の財務報告書などの文書が挙げられる．これら文書はしばしば，図表，グラフ，数式といった非テキスト要素を多く含むリッチドキュメントであり，これら要素を正確に扱うことが RAG システムの鍵となる [4, 5]．

リッチドキュメントを RAG で扱う代表的な方法として **Textual RAG** と **Visual RAG** がある．Textual RAG は，文書をテキスト形式へ変換し，そのテキストを索引化して検索する [4]．Textual RAG においてテキスト変換は特に重要であり，様々な手法が提案されている [6, 7, 8, 9, 10]．その中でも VLM によるテキスト変換は広範な文書が扱え，堅牢性においても利点があるため [5]，本研究では VLM を用いた

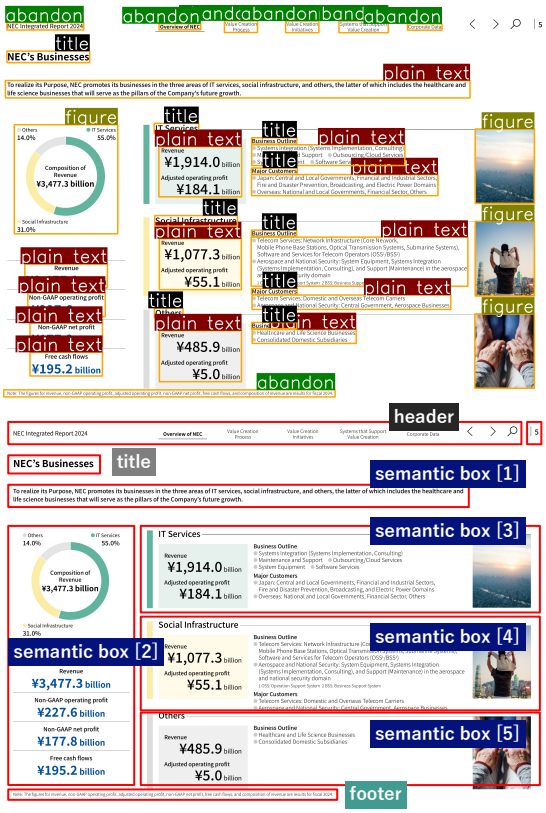

図 1 従来の文書レイアウト解析結果（上：DocLayout-YOLO）と提案手法（下：SCAN）の比較．

Textual RAG に注目する．Visual RAG は，ページやその領域を画像として索引化し，画像検索によって得た関連画像を VLM へ入力する [11, 12, 13]．

いずれのパイプラインも，視覚情報の理解においては VLM の読解能力に依存する．しかし，ページ全体を VLM で一度に処理しようとすると，情報過多による内容の欠落や文字の誤認識，ハルシネーションが発生しやすいという課題がある．

この課題に対し，ページを部分領域に分割するアプローチが考えられる．文書レイアウト解析は，文書ページからタイトル，段落，表，図，キャプションといった文書要素を検出する技術であり，OCR の前

* Equal contribution
† この研究は旧所属（NEC）在籍中の成果である．

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/).

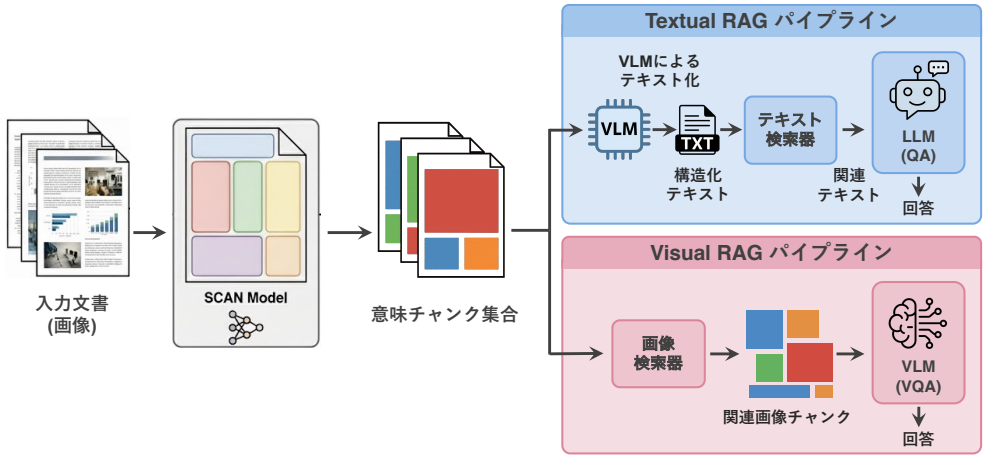

図 2　SCAN を既存の Textual RAG および Visual RAG パイプラインへ適用する際の全体概要.

処理や情報抽出に広く用いられている [14, 15, 16]. 図 1（上）に DocLayout-YOLO [14] による解析結果を示す．このような個々の文書要素を VLM の入力単位として利用すれば，情報過多による VLM の負荷を軽減できる可能性がある．しかし一方で，文書要素に基づく分割は同一トピック内の要素を断片化させ，図表とその説明文などの重要な文脈を分断するリスクがある．VLM は文脈を柔軟に解釈する能力を持つため，このような細かい粒度の分割は VLM にとって最適ではない．

本研究では，VLM による視覚情報の理解を最適化する**意味的文書レイアウト解析手法 SCAN**（**S**emanti**C** Document Layout **AN**alysis）を提案する．本手法では，文書要素ではなく単一のトピックによって統合されたより粗い領域を一つの**意味チャンク**として定義する．これにより，図 1（下）のように，関連する図表と本文が同一の矩形内に保持され，VLM は十分な文脈をもって内容を解釈できる．

2　提案手法

2.1　意味的文書レイアウト解析

本研究の目的は，リッチドキュメントの 1 ページを，VLM が扱いやすい部分領域に分割しつつ，図表と本文などの関連要素を同一領域内に保持することである（図 1）．このような領域を意味チャンクとよび，この問題を意味的文書レイアウト解析とよぶ．

本研究では意味チャンクを「一つのサブトピックに基づいて内容がまとまっている領域」と定義する．一般に，文書のページは一つのトピックについて述べられているが，複雑なページではその中に複数のサブトピックが含まれる．意味チャンクはこれらサブトピックのそれぞれに対応する領域である．図 1（下）では，*semantic box [3]* は IT Services に関するサブトピックに対応する．

実装上，意味チャンクは矩形領域として表現し，ページの内容を漏れや重複のないように分割する．意味チャンクは大きく 2 種類のラベルを持つ．**semantic box** は上述のように単一のサブトピックに対応する領域である．**global box** はタイトル，ヘッダ，フッタなどページ全体に関係する情報を含む領域である[1)]．本解析タスクは，入力を単一のページ画像とし，出力を $\{(\boldsymbol{b}_i, c_i)\}_{i=1}^{N}$（矩形 $\boldsymbol{b}_i$ とクラス c_i）とする多クラス物体検出として定式化される．

2.2　データセット構築

意味的文書レイアウト解析のためのデータセットを，CCpdf コーパス [17] の日本語サブセットから収集した文書画像を元に作成した．多様性を担保するため，MiniCPM Visual Embedding [18] で得た画像埋め込みに基づき階層的クラスタリングを行い，各クラスタの重心から近い距離のページを均等にサンプリングした．

画像へのタグ付けでは，まず作業者 3 名で試験的アノテーションを実施した．このとき，後述の Hungarian 法に基づく Intersection over Union (IoU) を用いて作業者間一致度を評価した．結果，0.7 以上の十分な一致度が得られたため，本番では 6 名かつ 1 サンプル 1 名体制で作業を進めた．最終的に 24,577 ページをタグ付けし，検証セットとテストセットにそれぞれ 1,000 ページを割り当てた．付録 A に詳細な作業指示や統計値を示す．

1)　global box はさらに *title* などのより詳細なラベルに分類される．

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/).

	OHR-Bench						BizMMRAG				Allganize			
	TXT	TAB	FOR	CHA	RO	ALL	TXT	CHA	TAB	ALL	TXT	CHA	TAB	ALL
分割なし	40.6	31.1	26.1	19.0	8.8	31.1	85.0	52.3	69.1	68.8	84.5	68.4	62.2	71.7
DiT	44.8	32.0	24.5	20.3	16.0	33.5 (+2.4)	75.0	54.6	50.0	59.9 (−8.9)	90.1	67.1	62.2	73.2 (+1.5)
DocLayout-YOLO	38.4	10.0	16.3	7.7	12.5	22.4 (−8.7)	61.7	25.0	28.6	38.4 (−30.4)	49.3	21.1	23.2	31.2 (−40.5)
Beehive	43.0	10.9	19.5	8.6	16.6	25.3 (−5.8)	70.0	29.6	21.4	40.3 (−28.5)	61.3	40.8	26.8	43.0 (−28.7)
SCAN（本手法）	44.4	31.9	26.6	20.6	17.7	**33.8** (+2.7)	81.7	72.7	73.8	**76.1** (+7.3)	85.9	85.5	72.0	**81.1** (+9.4)

表 1　Textual RAG の結果．質問は回答に必要な文書要素の種類（Text, Table, Formula, Chart, Reading Order）で分類されており，TXT, TAB, FOR, CHA, RO は各質問カテゴリを表す．ALL は各カテゴリのマイクロ平均である．

2.3　物体検出モデルの fine-tuning

事前学習済み物体検出器を fine-tuning して SCAN モデルを構築する．具体的には，CNN 系モデル [19, 20] と Transformer 系モデル [21, 22] の代表として，Ultralytics フレームワークで提供される YOLO11-X[2] および RT-DETR-X[3] を用いた．

推定された矩形の粒度と座標精度を同時に評価するため，マッチングに基づく IoU を用いる．推定矩形集合と正解矩形集合の要素間のマッチングを Hungarian 法 [23] で求め，対応の付いたペアの IoU を平均する．マッチしなかった矩形の IoU は 0 とし，検出数が過剰または過少だった場合にペナルティを与える．また，推定矩形集合の被覆率（推定矩形集合の和集合の面積 / 正解矩形集合の和集合の面積）も併用し，ハイパーパラメータを調整する．なお，検証において高い性能を示したことから，以後の実験では RT-DETR-X に基づく SCAN モデルを使用する．付録 C に SCAN モデルの出力例を示す．

2.4　Textual/Visual RAG への統合

SCAN は Textual RAG と Visual RAG の双方で前処理として用いる（図 2）．ページ画像に対して SCAN が意味チャンクを推定し，各矩形をクロップしてチャンク画像列を得る．Textual RAG では，各チャンク画像を VLM で Markdown 形式のテキストに変換し，矩形左上座標に基づく簡易的なルールで読み順を推定して連結し，ページ単位テキストとして索引化する．Visual RAG では，得られた画像列をそのまま画像チャンクとして検索対象とする．

3　実験

3.1　評価データセット

提案手法の Textual/Visual RAG における性能を，英語データセットである OHR-Bench [4] と，日本語データセットである BizMMRAG および Allganize [24] で評価した．いずれも，金融，公共，IT など様々なドメインの，RAG を必要とする質問応答タスクを含む．BizMMRAG は内製データセットである．

3.2　実験設定

各データセットにおいて，Textual RAG および Visual RAG の両手法を評価した．Textual RAG ではテキスト変換処理として，(1) レイアウト解析なしでページ全体をテキスト化，(2) 従来のレイアウト解析（DiT [15], DocLayout-YOLO [14], Beehive [16]）の後に各レイアウト要素をテキスト化，(3) SCAN でレイアウト解析の後に各レイアウト要素をテキスト化，を比較した．Visual RAG では検索用の画像チャンクの作り方として，(1) ページ全体を 1 チャンクとする設定，(2) 従来のレイアウト解析器（DiT, DocLayout-YOLO, Beehive）によりチャンクへ分割する設定，(3) SCAN により意味チャンクへ分割する設定，を比較した．

評価指標として，OHR-Bench [4] における Textual RAG では，回答と正解の文字列の F1 スコアを使用した．その他の設定では LLM-as-a-judge [25] により回答に 1–5 のスコアを付与し，4 以上を正解，それ以外を不正解として正解率を算出した．付録 B に各実験設定の詳細を示す．

3.3　Textual RAG の結果

OHR-Bench．表 1 の通り，SCAN によるページ分割により平均スコアは 31.1%から 33.8%へ改善した．一方，従来の細かい粒度のレイアウト解析は必ずしも有効ではなく，DocLayout-YOLO を組み合わせると平均スコアが 22.4%まで低下した．この低下は TAB や CHA カテゴリで顕著であり，過度な分割により図表周辺の関連情報が欠落したことが原因と考えられる．

BizMMRAG/Allganize．SCAN は両データセットで大幅な改善を達成した．多言語 VLM は英語で相

2）https://docs.ultralytics.com/models/yolo11/
3）https://docs.ultralytics.com/models/rtdetr/

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/).

	OHR-Bench						BizMMRAG				Allganize			
	TXT	TAB	FOR	CHA	RO	ALL	TXT	CHA	TAB	ALL	TXT	CHA	TAB	ALL
分割なし	84.0	68.6	71.5	58.7	67.9	70.2	71.7	56.8	57.1	58.9	75.9	71.1	62.5	69.9
DiT	80.7	63.8	62.2	51.9	66.0	64.9 (−5.3)	61.7	59.5	63.6	61.6 (+2.7)	81.8	68.8	64.1	71.6 (+1.7)
DocLayout-YOLO	72.2	57.9	58.3	47.5	62.4	59.6 (−10.6)	55.0	54.8	61.4	57.0 (−1.9)	69.5	68.8	65.8	68.0 (−1.9)
Beehive	73.2	60.1	64.2	43.8	87.6	65.8 (−4.4)	66.7	45.5	52.4	54.8 (−4.1)	66.0	52.6	60.0	59.5 (−10.4)
SCAN（本手法）	86.0	70.0	73.5	63.4	86.3	**75.8** (+5.6)	75.0	61.4	71.4	**69.3** (+10.4)	84.4	67.1	75.0	**75.5** (+5.6)

表 2　Visual RAG の結果.

分割手法	アーキテクチャ（パラメータ数）	ページあたりのチャンク数	相対サイズ (%)	Textual RAG スコア	Visual RAG スコア
分割なし	—	1.0	100.0	31.1	70.2
DiT	DiT (304M)	12.3	16.3	33.5	64.9
DocLayout-YOLO	YOLO11-X (57M)	9.9	11.3	22.4	59.6
Beehive	RT-DETR-L (43M)	17.4	4.8	25.3	65.8
SCAN$_{\text{YOLO}}$	YOLO11-X (57M)	3.2	26.4	33.5	72.4
SCAN$_{\text{RT-DETR}}$	RT-DETR-X (67M)	5.2	19.1	**33.8**	**75.8**

表 3　分割粒度と RAG 性能の関係．Textual/Visual RAG スコアは OHR-Bench の平均スコアである．

対的に高性能である一方，日本語等の非主要言語ではレイアウト・言語要因で劣化しやすい．SCAN は入力を意味的に自己完結な領域へ分解することで，この劣化を緩和し，日本語における大きな改善につながったと考えられる．

3.4　Visual RAG の結果

表 2 に示す通り，Visual RAG においても SCAN はレイアウト解析を行わない設定や従来のレイアウト解析を適用する設定を上回る性能を示した．特に，複数段落の対応付けを要する RO カテゴリで 18.4 ポイントと大きく改善した．これは，意味的文書レイアウト解析が，クエリに関係しない領域の混入を避けつつページ内の関連する段落を検索できることによる効果と考えられる．

3.5　分割粒度と RAG 性能の関係

分割粒度と RAG 性能の関係を定量化するため，OHR-Bench から無作為に 100 ページを抽出し，各手法のページあたりのチャンク数と相対サイズ（チャンク面積 / ページ面積）を算出した（表 3）．実験で用いた SCAN$_{\text{RT-DETR}}$ による分割は従来のレイアウト解析に比べ少数かつサイズが大きく，より粒度が粗いことがわかる．また，同一アノテーションで YOLO11-X を fine-tuning した SCAN$_{\text{YOLO}}$ は SCAN$_{\text{RT-DETR}}$ と同等の性能を示す．SCAN$_{\text{YOLO}}$ と同一のアーキテクチャを使用する DocLayout-YOLO のスコアが低いことから，改善の要因がアーキテクチャではなく我々の分割方針にあることが示唆される．

設定	トークン数（入力 + 出力）	チャンク数	時間（秒）
分割なし	1,320.4 + 991.9	1.0	68.0
SCAN	9,683.1 + 2,515.0	12.4	56.3

表 4　テキスト変換のトークン数および処理時間の比較．値は 1 ページあたりの平均値を示す．

3.6　テキスト変換コストの比較

Textual RAG において，ページを複数領域へ分割することで精度は向上するが，VLM へのリクエスト回数増加により処理時間や処理コストが増加する懸念がある．OHR-Bench から無作為に 10 ページを抽出し，`Qwen2.5-VL-72B` でテキスト変換する際のトークン数と処理時間を比較した（表 4）．SCAN は入力・出力トークン数を増加させた一方で，ページあたりの処理時間を 68.0 秒から 56.3 秒へ短縮した．これは，SCAN の分割により 1 リクエストあたりの入力トークン数が 1,320.4 から平均 780.9（= 9,683.1/12.4）へ減少し，attention 計算の負荷が下がったためと解釈できる．

4　おわりに

本研究では，Textual RAG および Visual RAG に適した意味的文書レイアウト解析手法 SCAN を提案した．SCAN は，ページを単一のトピックに基づいて統合された領域へ分割し，図表と関連する本文などの文脈情報を保持したまま VLM の処理負荷を軽減する．複数のデータセット・言語での評価により，SCAN は Textual RAG で 2.7–9.4 ポイント，Visual RAG で 5.6–10.4 ポイントの改善を達成した．今後は，文書要約，情報抽出，文書 VQA など，他の文書理解タスクへの適用が考えられる．

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/).

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, 2021.

[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] Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language models. In **Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining**, KDD '24, pp. 6491–6501, New York, NY, USA, 2024. Association for Computing Machinery.

[4] Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. Ocr hinders rag: Evaluating the cascading impact of ocr on retrieval-augmented generation. In **Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)**, pp. 17443–17453, October 2025.

[5] Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 24838–24848, June 2025.

[6] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system, 2022.

[7] Datalab. Marker. https://github.com/VikParuchuri/marker, 2024.

[8] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. **arXiv preprint arXiv:2409.18839**, 2024.

[9] 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, 2024.

[10] OpenAI. Gpt-4o system card. **arXiv preprint arXiv:2410.21276**, 2024.

[11] 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.

[12] Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Vdocrag: Retrieval-augmented generation over visually-rich documents. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, 2025.

[13] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models, 2024.

[14] Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception, 2024.

[15] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer. In **Proceedings of the 30th ACM International Conference on Multimedia**, MM '22, pp. 3530–3539, New York, NY, USA, 2022. Association for Computing Machinery.

[16] Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Nikolaos Livathinos, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Fabian Lindlbauer, Kasper Dinkla, Lokesh Mishra, Yusik Kim, Shubham Gupta, Rafael Teixeira de Lima, Valery Weber, Lucas Morin, Ingmar Meijer, Viktor Kuropiatnyk, and Peter W. J. Staar. Docling technical report, 2024.

[17] Michał Turski, Tomasz Stanisławek, Karol Kaczmarek, Paweł Dyda, and Filip Graliński. Ccpdf: Building a high quality corpus for visually rich documents from web crawl data. In **International Conference on Document Analysis and Recognition**, pp. 348–365. Springer, 2023.

[18] OpenBMB Rhapsody Group. Memex: Ocr-free visual document embedding model as your personal librarian. https://huggingface.co/RhapsodyAI/minicpm-visual-embedding-v0, 2024. Accessed: 2024-06-28.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In **Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)**, pp. 779–788, 2016.

[20] Rahima Khanam and Muhammad Hussain. Yolov11: An overview of the key architectural enhancements, 2024.

[21] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alex-ander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In **Proceedings of the European Conference on Computer Vision (ECCV)**, pp. 213–229. Springer, August 2020.

[22] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, pp. 16965–16974, June 2024.

[23] Harold W. Kuhn. The hungarian method for the assignment problem. **Naval Research Logistics Quarterly**, Vol. 2, pp. 83–97, 1955.

[24] Allganize.ai. Allganize rag leaderboard. https://huggingface.co/datasets/allganize/RAG-Evaluation-Dataset-JA, 2024.

[25] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A survey on llm-as-a-judge, 2025.

[26] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[27] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025.

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/).

A　データセット構築の詳細

アノテーションでは，まず各ページに対し global box を特定する．global box には，さらに *title*, *header*, *footer*, *date*, *author* の 5 種類のラベルを付与した．続いて，残りの領域をサブトピックに分割して semantic box を付与する．semantic box は以下の指針をもとに決定した．

- 可能な限り自己完結であること
- 領域の情報量が多すぎる場合は適度な粒度まで分割すること
- 矩形同士は非重複で，全体としてページ内容を取りこぼさず覆うこと
- サブトピックに対応する領域が矩形でなく，矩形状に覆うと他のサブトピックが混入する場合は，さらに分割して各矩形内のトピック一貫性を保つこと

表 5,6 にデータセットの統計値を示す．

クラスタのドメイン	ページ数	矩形数
Tables, Charts	3,766	17,024
Flyers, Magazines, Menus, Recipes	3,747	21,555
Maps, Traveling information	3,383	16,558
Itemized documents	3,771	19,062
Handwritten text	821	3,481
Vertical texts	3,768	16,520
Math	1,213	5,479
Manuals, Guidelines, Blueprints	4,108	23,472

表 5　データセットに含まれる文書ドメインの統計値．

矩形の種類	ページあたりの平均矩形数	矩形数
semantic box	3.13	76,862
title	0.49	12,137
header	0.38	9,416
footer	0.73	17,969
date	0.12	2,888
author	0.16	3,879
全種類	5.01	123,151

表 6　データセットに含まれる矩形数の統計値．

B　実験設定の詳細

B.1　Textual RAG

英語では OHR-Bench [4] に倣い，検索器に BGE-m3 [26] および BM25 を用い，生成器に `meta-llama/Llama-3.1-8B-Instruct` および `Qwen/Qwen2-7B-Instruct` を用いた．各クエリに対して上位 2 件の関連チャンクを検索し，生成器の回答を F1 で評価した．検索器と生成器の 4 通りの組み合わせの平均スコアを最終スコアとした．日本語では，検索器に `intfloat/multilingual-e5-large` を用い，上位 5 件の関連チャンクを `gpt-4o-2024-08-06` へ入力した．テキスト変換には `Qwen/Qwen2.5-VL-72B` [27] を用いた．

B.2　Visual RAG

Visual RAG では，いずれのデータセットにおいても画像埋め込みに ColQwen2-v1.0 [13] を用い，上位 5 件の関連画像チャンクを検索した．回答生成には `Qwen/Qwen2.5-VL-7B` を使用した．

C　意味的文書レイアウト解析の例

図 3,4 に我々の SCAN モデルの出力例を示す．図 3 から，SCAN モデルが適切に意味チャンクを特定できていることがわかる．一方で，図 4 では一部領域が重複して検出されている．従来の OCR タスクのように過不足なくテキストを抽出する用途では，こうした重複領域の除去が必要となる．

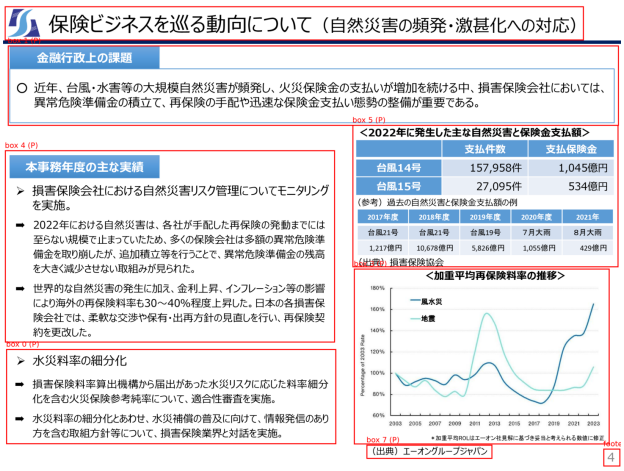

図 3　SCAN モデルの出力例．

図 4　SCAN モデルの出力例．

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/).