PDF 活用による日本語大規模マルチモーダルモデルの性能向上

白 定勳 [1]　相澤 彰子 [2]　相澤 清晴 [1]
[1] 東京大学　[2] 国立情報学研究所
{baek,aizawa}@hal.t.u-tokyo.ac.jp　aizawa@nii.ac.jp

概要

大規模マルチモーダルモデル（LMM）は英語では高い性能を示しているが，高品質な学習データの不足により，日本語における性能は依然として限定的である。既存の日本語 LMM は英語データの翻訳に依存することが多く，日本固有の文化的知識を十分に捉えられていない。本研究では，十分に活用されてこなかった日本語 PDF データに着目し，学習資源としての有効性を検討する。事前学習済みモデルを用い，レイアウト解析，OCR，および視覚と言語の対応付けにより，PDF から画像-テキスト対を自動抽出する完全自動化パイプラインを構築する。さらに，抽出した画像-テキスト対から指示データを生成し，学習データを拡充する。実験の結果，Heron-Bench において 2.1 %から 13.8 %の性能向上を達成し，PDF 由来データが日本語 LMM にとって有用なマルチモーダル資源であることを示した。

1　はじめに

大規模マルチモーダルモデル（LMM）は英語において高い性能を達成している一方，日本語では学習データの制約により性能が依然として限定的である [2, 3, 4]。近年，複数のオープンソース日本語 LMM が公開されているものの [5, 6, 7]，英語モデルとの差は依然として大きい。

英語では大規模な公開画像-テキスト対データセットが存在するのに対し，日本語 LMM は英語データの翻訳に依存する場合が多い [8, 6]。その結果，日本語固有の文化的知識を十分に学習できていないという課題がある。

本研究では，日本語 PDF データに着目し，文化的に関連性の高い知識を LMM 学習に取り込む。既存のマルチモーダルデータセットの多くが Web ベースであるのに対し [9, 10, 11]，PDF には書籍や各種文書に由来する有用な情報が含まれている。しかし，PDF データを日本語 LMM の学習に活用した研究は限られている。

図 1　PDF データの例. 日本語 LMM の学習に用いた多様な種類の PDF データを示す。

そこで本研究では，事前学習済みモデルを用いて，レイアウト解析，OCR，および視覚と言語の対応付けにより，PDF から画像-テキスト対を自動抽出する完全自動化パイプラインを構築する。さらに，抽出データから指示データを生成し，日本語 LMM を学習する。

実験の結果，Heron-Bench [6] において 2.1 %から 13.8 %の性能向上を達成し，PDF 由来データの有効性を確認した。本研究の主な貢献は以下のとおりである。

- PDF から画像-テキスト対を自動抽出する完全自動化パイプラインを構築し，手動アノテーションを不要とした。
- PDF 由来データにより，日本語 LMM の性能が Heron-Bench において 2.1 %から 13.8 %向上することを示した。
- 異なるモデルサイズ（3.8B，8B，14B）において，PDF 由来データの有効性を分析した。

本論文は ACL2025 Findings [1] で発表したものに準じている。

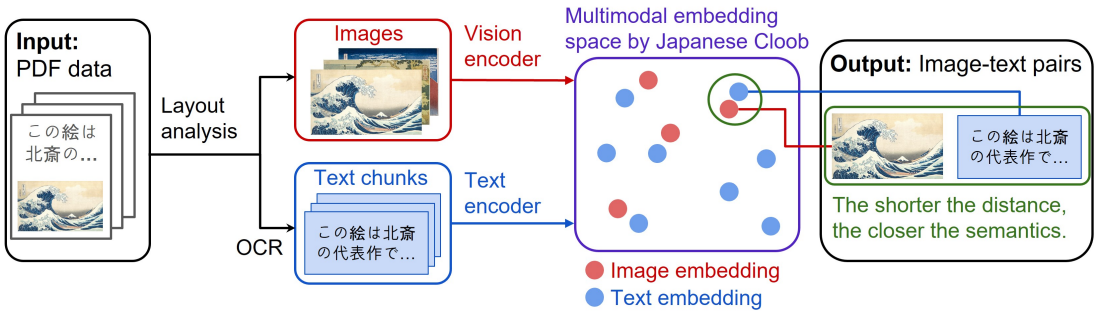

図 2　PDF から画像-テキスト対を抽出する自動化パイプライン. レイアウト解析，OCR，および視覚と言語の対応付けに事前学習済みモデルを用いる。

2　PDF データの活用

本節では，PDF データから日本語 LMM の学習データを構築する手法について述べる。

2.1　PDF データの収集

本研究では，国立国会図書館の Web ARchiving Project（NDL WARP）により提供された URL に基づき，Web 上から収集された PDF データを利用した [12]。収集された PDF は 5,138 万件を超えるが，後述する手順により一部を選別して使用する。これらの PDF には，学術論文に加え，ニュースレター，雑誌，報告書，広告，パンフレット，マニュアル，書籍など，多様な文書が含まれている。図 1 に，PDF データの例を示す。

2.2　画像-テキスト対の抽出

LMM の学習データを構築するため，本研究では PDF から画像-テキスト対を抽出する。全体の処理フローを図 2 に示す。

画像を含む PDF の選択. まず，画像を含まない PDF を除外する。多くの PDF は画像を含まず，含まれていても小さなロゴや記号のみである場合が多い。そこで，本研究では画像を含む可能性が高い PDF として，ページ数が 5 ページ以下のものを選択し，各 PDF の 1 ページ目のみを使用する。

PDF に画像が含まれているかの判定には，Python ライブラリである PyMuPDF [13] を用いる。その結果，200K 件の PDF（200K ページ）を選択した。

レイアウト解析および OCR による画像・テキスト抽出. PyMuPDF による直接抽出では，不可視画像の抽出や，1 つの画像が複数要素に分割される問題が生じる場合がある。これを防ぐため，本研究では PDF を JPEG 画像に変換した上で，画像およびテキストを抽出する。

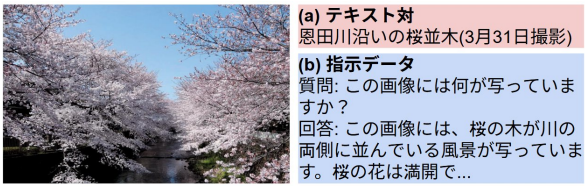

図 3　1 つの画像に対する（a）テキスト対 および（b）指示データの例.

抽出には，画像入力に基づく PDF 解析ツールである Surya [14] を使用する。Surya はレイアウト解析 [15] と OCR を実行し，画像領域とテキスト領域を特定した後，テキスト領域から文字情報を抽出する。これにより，画像とテキストの両方を取得する。Surya は 90 以上の言語に対応しており，日本語 PDF にも適用可能であるが，OCR の誤認識が生じる場合もある。

画像とテキストの対応付け. 各画像に対して，意味的に最も類似するテキストを対応付ける。画像および OCR で抽出したテキストをそれぞれエンコードし，コサイン類似度に基づいて対応付けを行う。対応付けには，事前学習済み視覚言語モデルである Japanese-Cloob [16, 17] を使用する。図 3(a) に，画像-テキスト対の例を示す。

2.3　指示データの生成

画像-テキスト対はそのまま LMM の学習に利用可能であるが，効果は限定的であった（補足資料 Table A 参照）。そこで，本研究では LLaVA [18] に倣い，GPT を用いて指示データを生成する。

具体的には，画像を GPT-4o-mini に入力し，指示データを生成する。対応付けられたテキストは文脈情報として利用し，プロンプトは日本語応答が得られるよう一部修正した。

実験の結果，対応テキストの品質が十分でない場合には，画像のみを用いて指示データを生成する方

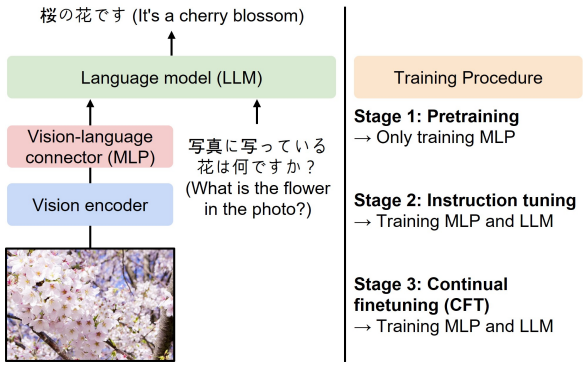

図 4　LMM の学習. 本研究では LLaVA1.5 [19] フレームワークを用い，事前学習，指示チューニング，および継続的ファインチューニング（CFT）の 3 段階で学習を行う。

Dataset	Stage	Count
LLaVA-Pretrain-JA	1	558K
LLaVA-v1.5-Instruct-620K-JA	2	620K
Instruct-from-200K PDF	3	362K

表 1　日本語 LMM の学習データセットの詳細.

が効果的であることが分かった（詳細は補足資料 §A.2 参照）。そのため，本研究では Table B を除き，すべての指示データを画像のみから生成している。200K 件の PDF から，合計 362K 件の指示チューニング用データを生成した。図 3(b) に例を示す。

3　日本語 LMM の学習

近年，複数のオープンソース日本語 LMM が公開されている [8, 5, 6, 7]。本研究では，PDF データの有効性を評価するため，LLaVA [18] フレームワークを採用し，LLaVA1.5 [19] を用いる。図 4 に，LLaVA1.5 のフレームワークを示す。

ハイパーパラメータの大部分は LLaVA1.5 の設定に従い，一部のみを変更した。具体的には，視覚エンコーダを CLIP（clip-vit-large-patch14-336）[20] から SigLIP（siglip-so400m-patch14-384）[21] に置き換え，Vicuna-7B [22] の代わりに複数の LLM を用いて実験を行う。LLM の選択については後述する。

学習には，パラメータ効率の高いファインチューニング手法である LoRA [23] を用いる。

3.1　学習手順

LLaVA1.5 の学習は，事前学習済みの視覚エンコーダと LLM を統合するため，複数の段階から構成されている。

ステージ 1：事前学習. 画像-テキスト対を用いて，視覚言語コネクタ（MLP）のみを学習する。

ステージ 2：指示チューニング. 視覚指示データを用い，MLP および LLM を指示チューニングする。

ステージ 3：継続的ファインチューニング（CFT）. 本研究では，ステージ 1 および 2 の後に，PDF 由来データを用いた継続的ファインチューニング（CFT）を追加する。この段階では，MLP および LLM の両方を学習する。

学習データ. 学習データの詳細を表 1 に示す。ステージ 1 および 2 では，日本語に翻訳された LLaVA 学習データ [6] を使用し，ステージ 1 では LLaVA-Pretrain-JA [24]（558K 件），ステージ 2 では LLaVA-v1.5-Instruct-620K-JA [25]（620K 件）を用いる。ステージ 3 では，§2 で構築した PDF 由来データ 362K 件を使用する。

3.2　LLM の選択

日本語 LLM. 日本語 LLM として，Llama3-8B を基盤とする Swallow（Llama-3-Swallow-8B-Instruct-v0.1）[26] を使用する。

汎用（非日本語）LLM. 日本語 PDF データの汎用性を検証するため，非日本語 LLM である Phi3-mini [27]（3.8B）および Phi3-medium [27]（14B）を使用する。

4　実験および分析

4.1　評価指標

日本語 LMM の評価には，標準的な日本語 LMM ベンチマークである Heron-Bench [6] を使用する。Heron-Bench は，102 問の日本語質問と 21 枚の日本固有画像から構成され，detail，conv，complex の 3 種類に分類されている。Heron-Bench は LLaVA-Bench [18] に基づく評価手法を採用している。まず GPT-4（`gpt-4-0125-preview`）[28] が参照解答を生成し，LLM-as-a-judge 手法 [29] により，LMM の出力と参照解答を評価する。最終スコアは，LMM の平均スコアを GPT-4 の平均スコアで正規化した比率（%）として算出される。

比較に用いた LMM. 比較対象として，GPT-4V [28]，Claude 3 Opus [30]，Gemini Pro [3] の 3 種類のプロプライエタリ LMM と，7 種類のオープンソース LMM を使用する。具体的なモデルは，LLaVA 1.6 7B [31], LLaVA 1.5 7B [18], Qwen-VL 7B [32], Japanese StableVLM 7B [8], EvoVLM-JP-v1-7B [5], Heron GIT [6] である。

	Method	Detail	Conv	Complex	Avg.
From Heron-Bench [6]	GPT-4V	83.3	77.5	78.3	79.7
	Claude 3 Opus	74.5	68.4	77.7	73.6
	Gemini Pro	55.6	64.3	64.0	61.3
	LLaVA 1.6 7B	30.9	37.3	31.0	33.1
	LLaVA 1.5 7B	42.4	45.9	35.5	41.3
	Qwen-VL 7B	46.3	50.6	52.3	49.7
	Japanese StableVLM 7B	25.2	51.2	37.8	38.1
	EvoVLM-JP-v1 7B	50.3	44.4	40.5	45.1
	Heron GIT 7B	42.8	54.2	43.5	46.8
Ours	LLaVA1.5-Swallow 8B	**70.1**	**62.3**	**65.0**	**65.8**
	LLaVA1.5-Phi3-mini 3.8B	61.6	61.2	48.5	57.1
	LLaVA1.5-Phi3-medium 14B	62.0	56.1	54.1	57.4

表 2　主結果. PDF 由来データを用いて学習した本研究のモデルは高い性能を示す。各モデル名はバックボーンとなる LLM に基づいて命名している（例：LLaVA1.5-Swallow）。

4.2　主結果

表 2 に，本研究で PDF 由来データを用いて学習したモデルの結果を示す。LLaVA1.5-Swallow は，Heron-Bench 論文で報告されている既存のオープンソース日本語 LMM を上回る性能を達成した。特に，従来の最高平均スコア 49.7%に対し，16.1%の改善が確認された。また，LLaVA1.5-Phi3-mini および LLaVA1.5-Phi3-medium も既存モデルを上回る性能を示した。これらの結果から，PDF データの活用が日本語 LMM の性能向上に有効であることが分かる。

4.3　PDF 由来データは有効か

表 3 に，各 LLM に対する LLaVA1.5 の学習結果を学習ステージごとに示す。ステージ 1（事前学習）直後では性能は低いが，ステージ 2（指示チューニング）後に Heron-Bench の性能が約 20%〜30%向上する。さらに，ステージ 3（PDF データを用いた継続的ファインチューニング）により，2.1%（Phi3-medium）から 13.8%（Phi3-mini）の追加的な性能向上が得られた。これらの結果から，PDF 由来データを用いた CFT が有効であることが分かる。

ステージ 3 では，PDF データ量を段階的に増加させた実験も行った。PDF を 50K 件追加するごとに約 90K 件の指示データが生成され，性能向上が確認されたが，その効果はデータ量に対して必ずしも線形にはスケールしない。

また，日本語 PDF データは，非日本語 LLM を日本語 LMM へ適応させる上でも有効である。Phi3-mini および Phi3-medium を用いた実験（§3.2）においても，PDF 由来データによる性能向上が確認された。この効果は Swallow 8B に限らず，3.8B および 14B といった異なるモデルサイズにおいて一貫して観測された。

LLM	Stage	Detail	Conv	Complex	Avg.
Swallow 8B	1. Pretraining	23.3	23.9	20.6	22.6
	2. Instruction tuning	54.0	50.5	59.6	54.7
	3. CFT on 50K PDF	67.3	**69.0**	60.9	65.7
	3. CFT on 100K PDF	**70.1**	62.3	**65.0**	**65.8**
	3. CFT on 150K PDF	66.7	61.6	63.0	63.8
	3. CFT on 200K PDF	65.6	63.7	64.6	64.6
Phi3-mini 3.8B	1. Pretraining	20.7	18.2	16.2	18.4
	2. Instruction tuning	45.8	43.1	40.9	43.3
	3. CFT on 50K PDF	60.2	54.9	40.6	51.9
	3. CFT on 100K PDF	56.6	55.3	47.1	53.0
	3. CFT on 150K PDF	61.6	**61.2**	**48.5**	**57.1**
	3. CFT on 200K PDF	**62.0**	54.3	46.5	54.3
Phi3-med. 14B	1. Pretraining	25.8	26.2	18.3	23.4
	2. Instruction tuning	56.4	50.1	**56.0**	54.2
	3. CFT on 50K PDF	**66.5**	**59.2**	50.6	**58.8**
	3. CFT on 100K PDF	65.2	52.6	51.1	56.3
	3. CFT on 150K PDF	62.0	56.1	54.1	57.4
	3. CFT on 200K PDF	**66.5**	54.5	53.3	58.1

表 3　各学習ステージおよび PDF データ量ごとの結果.

5　結論

本研究では，日本語 PDF データを活用した LMM 学習手法を検討し，画像-テキスト対を自動抽出する完全自動化パイプラインを構築した。実験の結果，PDF 由来データを学習に取り入れることで，Heron-Bench において最大 13.8%の性能向上を達成した。さらに，PDF 由来データは異なるモデルサイズにおいても有効であり，既存のマルチモーダルデータセットを補完する有用な学習資源であることを示した。

謝辞

本研究では，国立情報学研究所大規模言語モデル研究開発センターより PDF データセットの提供を受けた。特に，著作権および利用許諾に関してご助言いただいた河原大輔教授に感謝する。本研究は，JSPS KAKENHI（課題番号 24K23882）および NVIDIA Academic Grant Program の支援を受けた。

参考文献

[1] Jeonghun Baek, Akiko Aizawa, and Kiyoharu Aizawa. Harnessing PDF data for improving Japanese large multimodal models. In **ACL 2025 Findings**, 2025.

[2] OpenAI. Gpt-4o, 2024. Accessed: 2025-02-15.

[3] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. **arXiv preprint arXiv:2403.05530**, 2024.

[4] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. **arXiv:2407.21783**, 2024.

[5] Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. **Nature Machine Intelligence**, 2025.

[6] Yuichi Inoue, Kento Sasaki, Yuma Ochi, Kazuki Fujii, Kotaro Tanahashi, and Yu Yamaguchi. Heron-bench: A benchmark for evaluating vision language models in japanese. In **CVPR The 3rd Workshop on Computer Vision in the Wild**, 2024.

[7] Keito Sasagawa, Koki Maeda, Issa Sugiura, Shuhei Kurita, Naoaki Okazaki, and Daisuke Kawahara. Constructing multimodal datasets from scratch for rapid development of a japanese visual language model. **arXiv preprint arXiv:2410.22736**, 2024.

[8] Makoto Shing and Takuya Akiba. Japanese stable vlm, 2023.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In **ECCV**, 2014.

[10] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In **ACL**, 2018.

[11] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. **NeurIPS**, 2022.

[12] National Diet Library. Web Archiving Project (WARP). Accessed: 2025-02-18.

[13] Artifex Software Inc. PyMuPDF (pymupdf). Accessed: February 15, 2025.

[14] Vik Paruchuri. Surya: A pdf processing library, 2024. Accessed: February 15, 2025.

[15] Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. Layoutparser: A unified toolkit for deep learning based document image analysis. In **ICDAR**, 2021.

[16] Makoto Shing, Tianyu Zhao, and Kei Sawada. rinna/japanese-cloob-vit-b-16, 2022. Accessed: February 15, 2025.

[17] Kei Sawada, Tianyu Zhao, Makoto Shing, Kentaro Mitsui, Akio Kaga, Yukiya Hono, Toshiaki Wakatsuki, and Koh Mitsuda. Release of pre-trained models for the Japanese language. In **LREC-COLING 2024**, 2024.

[18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In **NeurIPS**, 2023.

[19] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In **CVPR**, 2024.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In **ICML**, 2021.

[21] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In **ICCV**, 2023.

[22] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. Accessed: 2025-02-15.

[23] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In **ICLR**, 2022.

[24] Turing Motors. Llava-pretrain-ja. https://huggingface.co/datasets/turing-motors/LLaVA-Pretrain-JA, 2024. Accessed: February 15, 2025.

[25] Turing Motors. Llava-v1.5-instruct-620k-ja. https://huggingface.co/datasets/turing-motors/LLaVA-v1.5-Instruct-620K-JA, 2024. Accessed: February 15, 2025.

[26] Naoaki Okazaki, Kakeru Hattori, Hirai Shota, Hiroki Iida, Masanari Ohi, Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Rio Yokota, and Sakae Mizuki. Building a large japanese web corpus for large language models. In **Proceedings of the First Conference on Language Modeling**, 2024.

[27] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. **arXiv:2404.14219**, 2024.

[28] OpenAI. GPT-4 Technical Report, 2023. Accessed: 2025-02-15.

[29] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In **NeurIPS**, 2024.

[30] Anthropic. Claude 3 family, 2024. Accessed: 2025-02-15.

[31] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024.

[32] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. **arXiv preprint arXiv:2308.12966**, 2023.

Method	Detail	Conv	Complex	Avg.
Stages 1 and 2	54.0	**50.5**	**59.6**	**54.7**
Top 1	37.6	40.7	40.4	40.0
Top 3	25.6	39.8	33.6	34.7
Top 5	25.0	19.5	25.3	22.8
Neighbor	**55.7**	45.4	43.1	46.4

表 4　未加工の画像-テキスト対を用いた学習結果. 画像-テキスト対のみを用いて CFT を行った場合，全体的にベースラインを下回る性能となる。

A　追加実験

A.1　指示データを生成せず，未加工の画像-テキスト対を用いた場合

抽出した画像-テキスト対から指示データを生成する代わりに，それらをそのまま用いて LMM を学習することも可能である。本実験では，50K 件の PDF データから抽出した画像-テキスト対を用いて，LLaVA1.5-Swallow を学習した。表 4 にその結果を示す。

結果から，ステージ 1 および 2 まで学習した LLaVA1.5 と比較して，性能が全体的に低下することが分かる。表中の Top 1，Top 3，Top 5 は，画像との対応付けにおいて，コサイン類似度が上位 1 件，3 件，5 件のテキストを用いた場合の結果を示している。Neighbor は，Top 1 のテキストに加え，同一 PDF 内の直前および直後のテキストを含めた設定を表す。Top 3，Top 5，あるいは Neighbor を用いた場合でも，ステージ 2 まで学習した LLaVA1.5 の性能を下回っている。この結果は，事前学習済みモデルのみによって抽出された画像-テキスト対が，学習データとして十分に有効ではないことを示唆している。

この性能低下には，主に 2 つの要因が考えられる。(1) 多くの PDF には，画像を直接説明するテキストがそもそもほとんど含まれていない点である。(2) 事前学習済みモデルの性能的制約である。レイアウト解析，OCR，および画像-テキスト対応付けの精度が十分でなかったため，画像-テキスト対の品質が低下した可能性がある。これらの PDF には正解アノテーションが存在しないため，どの要因が主要因であるかを厳密に特定することは困難である。しかし，数百件の例を手動で確認した結果，OCR により抽出されたテキストが不正確であるケースが多く観察された。具体的には，不要な改行が挿入されることで単語や文が分断されたり，複雑な漢字が誤認識される例が頻繁に見られた。

Data source for instruction	Detail	Conv	Complex	Avg.
Image	**67.3**	**69.0**	**60.9**	**65.7**
Image and paired text	65.9	65.7	60.2	63.9

表 5　指示データ生成におけるデータソースの比較. 本実験では LLaVA1.5-Swallow を使用する。

A.2　指示データ生成において対応テキストは有効か

指示データ生成時には，画像に対応付けられたテキストをコンテキストとして使用している。しかし，PDF から抽出されたテキストは不完全でノイズを含む場合が多く，その有効性は明らかではない。そこで本実験では，50K 件の PDF から得られたデータを用い，異なる情報源に基づいて指示データを生成し，その結果を表 5 に示す。

具体的には，画像のみを用いた場合と，画像に対応テキストを付加した場合の 2 条件を比較した。実験の結果，画像のみを用いて生成した指示データの方が高い性能を示した。ノイズを含む対応テキストは，指示データ生成において有効ではないことが分かる。この結果は，PDF から画像情報のみを抽出する場合でも十分に有用であり，対応テキストの品質が低い場合には，画像のみを用いる方が望ましいことを示唆している。そのため，200K 件の PDF を用いた実験では，すべて画像のみを用いて指示データを生成した。