Multimodal Structured Generation:
CVPR's 2nd MMFM Challenge Technical Report

Franz Louis Cesista
franzlouiscesista@gmail.com

June 17, 2024

Abstract

Multimodal Foundation Models (MMFMs) have shown remarkable performance on various computer vision and natural language processing tasks. However, their performance on particular tasks such as document understanding is still limited. They also require more compute, time, and engineering resources to finetune and deploy compared to traditional, unimodal models. In this report, we present Multimodal Structured Generation, a general framework which constrains the output logits of frozen MMFMs to force them to reason before responding with structured outputs that downstream APIs can parse and use. We provide a detailed account of our approach, including the technical details, theoretical discussions, and final evaluation results in the 2nd Multimodal Foundation Models Challenge hosted by the Computer Vision and Pattern Recognition (CVPR) conference. Our approach achieved the second highest score in the hidden test set for Phase 2 and third highest overall. This shows the method's ability to generalize to unseen tasks. And that simple engineering can beat expensive & complicated modelling steps as we first discussed in our paper, Retrieval Augmented Structured Generation: Business Document Information Extraction as Tool Use.[1]

1 Introduction

Multimodal Foundation Models (MMFMs) have been made possible by recent advances on grafting together parts from different foundation models pretrained on specific modalities [1]. The resulting "Frankenstein" models show remarkable results in diverse & multimodal tasks. However, their performance on document understanding is still lacking.

In this report, we present Multimodal Structured Generation, a general framework for controlling the output format of multimodal models. In this challenge, in particular, we use it to force frozen multimodal models to reason before responding with a structured output that downstream APIs can parse and use.

Our team actually only learned about the challenge roughly two days before the submissions deadline–the first 24 hours of which was wasted working with a commercially-available model which, after clarifying with the organizers, we were not allowed to use for the challenge. We neither had the time, compute resources, nor the budget to implement complicated modelling steps. Fortunately, we have already shown in our previous work on Retrieval Augmented Structured Generation (RASG) that simple engineering could supplant more complicated modelling steps–at least on document information extraction tasks [2]. RASG has four components: (1) Structured Generation [3] (2) Retrieval Augmented Generation [4], (3) Supervised Finetuning, & (4) Structured Prompting [5]. But for this challenge, we only managed to implement Structured Generation but with multimodal models instead of just an LLM as in our previous work.

Our team placed 2nd in Phase 2 of CVPR's 2nd MMFM Challenge and 3rd in the overall rankings–beating multiple teams that have finetuned their own multimodal models. Phase 2 contains never-before-seen evaluation datasets. This shows the generality of the approach to unseen tasks. And it being finetuning-free makes it easy and cheap for other teams to replicate. All of our scripts and evaluation results can be accessed in https://github.com/leloykun/MMFM-Challenge.

2 Methodology

2.1 Multimodal Structured Generation

Outputs from generative models are not guaranteed to be parseable by downstream programs, compilers, and/or APIs. For example, if a human user asks an LLM to generate a Python script for sorting a list of

[1] All of our scripts, deployment steps, and evaluation results can be accessed in https://github.com/leloykun/MMFM-Challenge

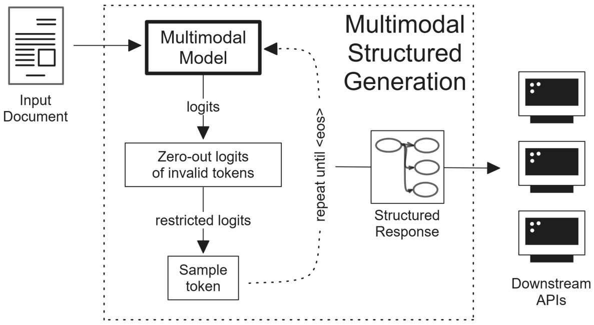


Figure 1: Multimodal Structured Generation

numbers, the LLM's output may *seem* runnable but in fact produce errors when ran on an IDE. There is a spectrum of approaches that guarantee the generative models' outputs can indeed be usable by downstream systems. On one end are what we call "soft" constraints which simply ask or instruct the models to follow a specified schema and retry until they succeed [6]. On the other end are "hard" constraints which zero-out the logits of invalid tokens altogether [3].

Structured Generation can then easily be applied to generative Multimodal Models as the latter also produce logits which we can then zero-out if they correspond to invalid tokens. See Figure 1.

2.2 Implementation Details

We used Huggingface's Inference Endpoints API to deploy the multimodal models and Huggingface's Text Generation Interface (TGI) API to call inference requests on the models [7] [8].

For Phase 1, since the test dataset is already publicly-available, we decided not to focus on it as the results would not show the generality of our approach. Thus, we only used an unaugmented Llava v1.5 for Phase 1 [9]. For Phase 2, we used Llava-Next (v1.6) augmented with Structured Generation to generate the results for the `mychart` and `myinfographic` datasets [10] [11]. But to maximize results, we used a version of Nous Hermes 2 Pro - Mistral 7B model augmented with Structured Generation for the `mydoc` dataset [12]. We finetuned this version of Hermes 2 Pro on the DocILE dataset [13] not in *this* challenge, but in our previous work [2]. This model can be downloaded from https://huggingface.co/leloy/Nous-Hermes-2-Pro-Docile-RASG-1ShotRetrieval-StructuredPrompt.

We used Structured Generation to force the multimodal models to reason *before* answering. We used slightly different *json* output formats for each of the evaluation datasets (which can be seen in our repository), but they generally follow the following template:

```
{
    "name": "<tool␣name␣e.g.␣infographic_explair_tool>",
    "description": "<tool␣description␣e.g.␣Infographic␣Explainer␣Tool>",
    "parameters": {
        "type": "object",
        "properties": {
            "1_reasoning": {"type": "string"},
            "2_answer": {
                "type": "string",
                "description": "Concise␣answer␣to␣the␣user␣question."
            },
        },
        "required": ["1_reasoning", "2_answer"],
    },
}
```

For `mydoc`, we used the entity being requested as the "key" following the json format we used in our previous work [2]. This makes it easy to request for multiple entities in the document (e.g. requesting for the `Billing Name` and `Total Amount` at the same time).

```
{
    "name": "doc_extraction_tool",
    "description": "Extract␣information␣from␣a␣document",
    "parameters": {
        "type": "object",
        "properties": {
            "1_reasoning": {"type": "string"},
            f"2_{key}": {
                "type": "integer" if key == "page" else "string",
                "description": "The␣answer,␣exactly␣as␣it␣appears␣in␣the␣document.",
                "maxLength": max_length,
            }
        },
        "required": ["1_reasoning", f"2_{key}"],
    },
}
```

Note that we prepended indices to the keys in the json format. That is because the version of TGI we used still uses an older version of Outlines (version $< 0.40.0$) which implicitly reorders the keys by alphabetical order. Prepending the indices solves this issue. But they can be removed on later versions of TGI and Outlines.

Also note that we asked the models to output the *exact* answers for `mydoc` while we only asked the models to be concise for `mychart` and `myinfographic`. This is because we suspect that the challenge organizers used the MMMU evaluation metric for `mydoc` and Mistral 7B to judge the outputs for `mychart` and `myinfographic` [14] [15]. The former requires exact outputs while implementation details on the evaluation script for the latter indicate that concise answers would be better. Although, these claims are unconfirmed.

2.3 Datasets

The MMFM challenge provided the following datasets for Phase 1: IconQA [16], FUNSD [17], WildReceipt [18], TextbookQA [19], TabFact [20], DocVQA [21], InfographicVQA [22], WebSRC [23], and WTQ [24].

3 Results

We evaluated our approach on the two phases of CVPR's 2nd MMFM's Challenge.

Table 1: Evaluation results on CVPR's 2nd MMFM Challenge

Phase	Eval Dataset	Acc
1	`iconqa_fill`	15.5%
	`funsd`	32.5%
	`iconqa_choose`	31.0%
	`wildreceipt`	35.5%
	`textbookqa`	51.5%
	`tabfact`	48.5%
	`docvqa`	20.5%
	`infographicvqa`	23%
	`websrc`	28.5%
	`wtq`	8.5%
	Phase 1 Overall	**29.5%**
2	`mydoc`	62.25%
	`mychart`	4.5%
	`myinfographic`	60.98%
	Phase 2 Overall	**50.49%**

Our approach placed 2nd in the hidden test set of Phase 2 and 3rd place overall.

4 Discussion

Interestingly, we managed to beat multiple teams who finetuned multimodal (vision + text) models using just an LLM and & structured generation on the Key-Information Extraction dataset, MYDOC. This supports one finding in our paper that visual information isn't really important for KIE–and in this particular case, using vision encoders was even harmful [2].

We have four hypotheses on why:

Table 2: Ablation Benchmarks of RASG components on KIE & LIR tasks on the DocILE dataset

Model	**Key-Information Extraction** **F1 Score**	**Line Items Recognition** **GLIRM-F1** [2]
GPT-3.5	34.17%	28.31%
+ 1-Shot Retrieval	+ 22.08%	+ 20.67%
+ Supervised Finetuning	+ 22.31%	+ 17.73%
+ Structured Prompting	+ 4.96%	+ 19.42%
Hermes 2 Pro - Mistral 7B	13.55%	4.69%
+ 1-Shot Retrieval	+ 36.87%	+ 40.55%
+ Supervised Finetuning	+ 17.71%	+ 13.53%
+ Structured Prompting	+ 0.63%	+ 10.30%

* Benchmarks results ablating three components of Retrieval Augmented Structured Generation on Key-Information Extraction (KIE) & Line Items Recognition (LIR) tasks on the DocILE dataset [13]: (1) Retrieval Augmented Generation [4], (2) Supervised Finetuning, & (3) Structured Prompting [5]. Structured Generation was not included in the ablation benchmarks as it is a necessary component of RASG to ensure that the outputs are parseable by downstream APIs [3]. Results show that adding Structured Prompting, i.e. infusing layout information to the text prompt, only adds a marginal increase in performance.

- First, perhaps **the visual and layout information are not important for Key-Information Extraction**.

  The team behind DocLLM had this idea of removing the vision component and treating the layout information as its own modality. And their Text + Layout only model worked just as well or even better than the Text + Vision and Text + Vision + Layout models they benchmarked [25]. Our previous work, on the other hand, completely gets rid of the other modalities and replaces them with other augmentations (i.e. retrieval augmented generation, structured generation, infusing the layout information to the prompt, & finetuning) instead [2]. There, we found that infusing the layout information to the text prompt does not actually help for the KIE task either. See Table 2.

  We also do not expect randomly permuting the order of the words in the text prompt would help either. So perhaps what is actually important for the KIE task is *locality*. That is, that nearby words in the image has to be nearby in the text prompt. But this is already guaranteed by good OCRs.

- Second, perhaps **the LLMs can already infer the location of the words in the image from the position (index) of the words in the text prompts**.

  Previous work show that LLMs trained without positional encodings can still learn position information [26]. The possible reason is that they infer this from the (implicit) causal graph. Perhaps this is also true in the 2D case. That is, we do not need to feed information on where the input words are because the LLMs can already infer this from the order of the words as they are fed to the LLM.

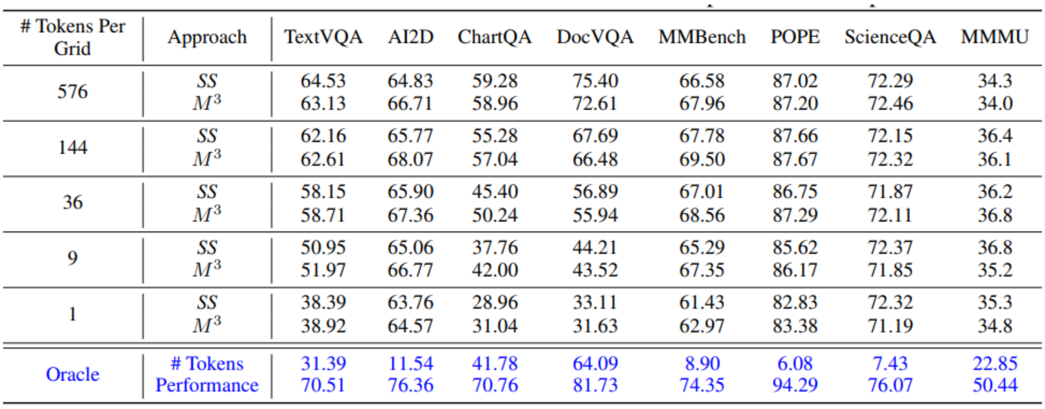
- Third, perhaps **it has to do with the information capacity of these models**.

  The base LLM of the LLaVA-NeXT (v1.6) model we tested is Mistral-7B [10]. And from a quick napkin math with the neural scaling laws, this indicates that the language model can only hold around 7*20 = 140B tokens of information [27]. This is not a lot. And then we further pushed it to overcapacity by jamming in the embeddings from the visual encoder and projector.

- Fourth, which we find most compelling, is that **we are simply not using enough image tokens for document information understanding**.

  Unlike pictures of pets (which only require a couple of tokens to describe), document images are **packed** with information. And from empirical observations, business documents usually have around $3,000$ text tokens. And we would have more tokens if we also consider the layout information. And yet, we usually pack them to image encoders that compresses the document to just $\leq$ 1024 tokens. That's a lot of information lost.

  Previous work which aims to reduce the number of image tokens shows empirical evidence for these claims [28]. In the table in Figure 2, see that more image tokens are required to achieve maximum performance on document understanding datasets.



# Tokens Per Grid	Approach	TextVQA	AI2D	ChartQA	DocVQA	MMBench	POPE	ScienceQA	MMMU
576	*SS*	64.53	64.83	59.28	75.40	66.58	87.02	72.29	34.3
	M^3	63.13	66.71	58.96	72.61	67.96	87.20	72.46	34.0
144	*SS*	62.16	65.77	55.28	67.69	67.78	87.66	72.15	36.4
	M^3	62.61	68.07	57.04	66.48	69.50	87.67	72.32	36.1
36	*SS*	58.15	65.90	45.40	56.89	67.01	86.75	71.87	36.2
	M^3	58.71	67.36	50.24	55.94	68.56	87.29	72.11	36.8
9	*SS*	50.95	65.06	37.76	44.21	65.29	85.62	72.37	36.8
	M^3	51.97	66.77	42.00	43.52	67.35	86.17	71.85	35.2
1	*SS*	38.39	63.76	28.96	33.11	61.43	82.83	72.32	35.3
	M^3	38.92	64.57	31.04	31.63	62.97	83.38	71.19	34.8
Oracle	# Tokens	31.39	11.54	41.78	64.09	8.90	6.08	7.43	22.85
	Performance	70.51	76.36	70.76	81.73	74.35	94.29	76.07	50.44

Figure 2: Comparison of approaches with the SS baseline and Matryoshka Multimodal Models (M^3) across various benchmarks under LLaVA-NeXT [28]. Here # Tokens denotes the number of visual tokens per image grid in LLaVA-NeXT. SS denotes the baseline model trained with a Specific Scale of visual tokens. M^3 is at least as good as SS, while performing better on tasks such as TextVQA, ChartQA, and MMBench. Oracle denotes the case where the best tradeoff between visual tokens and performance is picked.

5 Conclusion

Our results and final standings in CVPR's 2nd MMFM Challenge show our approach's ability to generalize to unseen tasks. While our implementation for this challenge is finetuning-free (which also makes it easy and cheap to implement for other teams), we could also apply it on finetuned models. However, as we noted in our previous work, naively combining finetuning and Structured Generation can lead to worse results [2]. Future work on the topic is necessary. But for now, ensuring that the alignment of output formats in the finetuning dataset and in the inference requests helps.

Our results on the mydoc dataset also reinforce our claim that neither the visual nor precise layout information is necessary for solving key-information extraction. Although more work is required to verify the four hypotheses we raised in the discussion section above.

References

[1] F. Bordes, R. Y. Pang, A. Ajay, *et al.*, *An introduction to vision-language modeling*, 2024. arXiv: 2405.17247 [cs.LG].

[2] F. L. Cesista, R. Aguiar, J. Kim, and P. Acilo, *Retrieval Augmented Structured Generation: Business Document Information Extraction as Tool Use*, 2024. arXiv: 2405.20245 [cs.CL].

[3] B. T. Willard and R. Louf, *Efficient guided generation for large language models*, 2023. arXiv: 2307.09702 [cs.CL].

[4] O. Ram, Y. Levine, I. Dalmedigos, *et al.*, "In-context retrieval-augmented language models," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1316–1331, 2023, ISSN: 2307-387X. DOI: 10.1162/tacl_a_00605.

[5] W. Wang, Y. Li, Y. Ou, and Y. Zhang, *Layout and task aware instruction prompt for zero-shot document image question answering*, 2023. arXiv: 2306.00526 [cs.CL].

[6] J. Liu, *Instructor*, https://python.useinstructor.com/ [Accessed: 2024-06-09], 2023.

[7] Huggingface Team, *Hugginface Inference Endpoints documentation*, https://huggingface.co/docs/inference-endpoints/index [Accessed: 2024-06-09], 2024.

[8] Huggingface Team, *Hugginface Text Generation Interface documentation*, https://huggingface.co/docs/text-generation-inference/index [Accessed: 2024-06-09], 2024.

[9] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, *et al.*, Eds., vol. 36, Curran Associates, Inc., 2023, pp. 34 892–34 916. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

[10] H. Liu, C. Li, Y. Li, *et al.*, *LLaVA-NeXT: Improved reasoning, OCR, and world knowledge*, Jan. 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-01-30-llava-next/.

[11] H. Liu, C. Li, Y. Li, and Y. J. Lee, *Improved baselines with visual instruction tuning*, 2023. arXiv: 2310.03744.

[12] interstellarninja, Teknium, theemozilla, karan4d, and huemin_art, *Hermes 2 Pro - Mistral 7B*, https://huggingface.co/NousResearch/Hermes-2-Pro-Mistral-7B [Accessed: 2024-06-09], 2024.

[13] Š. Šimsa, M. Šulc, M. Uřičář, *et al.*, *DocILE benchmark for document information localization and extraction*, 2023. arXiv: 2302.05658 [cs.CL].

[14] X. Yue, Y. Ni, K. Zhang, *et al.*, "MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI," in *Proceedings of CVPR*, 2024.

[15] Mistral Team, *Mistral 7B*, https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1 [Accessed: 2024-06-09], 2024.

[16] P. Lu, L. Qiu, J. Chen, *et al.*, "IconQA: A new benchmark for abstract diagram understanding and visual language reasoning," in *The 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks*, 2021.

[17] G. Jaume, H. Kemal Ekenel, and J.-P. Thiran, "FUNSD: A dataset for form understanding in noisy scanned documents," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, IEEE, Sep. 2019. DOI: 10.1109/icdarw.2019.10029.

[18] H. Sun, Z. Kuang, X. Yue, C. Lin, and W. Zhang, *Spatial dual-modality graph reasoning for key information extraction*, 2021. arXiv: 2103.14470 [cs.CV].

[19] D. Kim, S. Kim, and N. Kwak, "Textbook question answering with multi-modal context graph understanding and self-supervised open-set comprehension," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3568–3584. DOI: 10.18653/v1/P19-1347. [Online]. Available: https://aclanthology.org/P19-1347.

[20] W. Chen, H. Wang, J. Chen, *et al.*, "TabFact: A large-scale dataset for table-based fact verification," in *International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.

[21] M. Mathew, D. Karatzas, and C. V. Jawahar, "DocVQA: A dataset for VQA on document images," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2021. DOI: 10.1109/wacv48630.2021.00225.

[22] M. Mathew, V. Bagal, R. Tito, *et al.*, "InfographicVQA," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2022. DOI: 10.1109/wacv51458.2022.00264. [Online]. Available: http://dx.doi.org/10.1109/WACV51458.2022.00264.

[23] X. Chen, Z. Zhao, L. Chen, *et al.*, "WebSRC: A dataset for web-based structural reading comprehension," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2021. DOI: 10.18653/v1/2021.emnlp-main.343. [Online]. Available: http://dx.doi.org/10.18653/v1/2021.emnlp-main.343.

[24] A. Drozdov, N. Schärli, E. Akyürek, *et al.*, *Compositional semantic parsing with large language models*, 2022. arXiv: 2209.15003 [cs.CL].

[25] D. Wang, N. Raman, M. Sibue, *et al.*, *DocLLM: A layout-aware generative language model for multimodal document understanding*, 2023. arXiv: 2401.00908 [cs.CL].

[26] A. Haviv, O. Ram, O. Press, P. Izsak, and O. Levy, "Transformer language models without positional encodings still learn positional information," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 1382–1390. DOI: 10.18653/v1/2022.findings-emnlp.99. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.99.

[27] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, *Scaling laws for neural language models*, 2020. arXiv: 2001.08361 [cs.LG].

[28] M. Cai, J. Yang, J. Gao, and Y. J. Lee, *Matryoshka multimodal models*, 2024. arXiv: 2405.17430.