Breaking the annotation barrier with DocuLite: A scalable and privacy-preserving framework for financial document understanding

Saiful Haq[1,2], **Daman Deep Singh**[1], **Akshata Bhat**[1],
Krishna Chaitanya[1], **Prashant Khatri**[1], **Abdullah Nizami**[1], **Abhay Kaushik**[1]
Niyati Chhaya[1], **Piyush Pandey**[2]
Hyperbots Inc[1] IIT Bombay[2]

Abstract

In this paper, we introduce Doculite, a scalable and privacy-preserving framework for adapting large language models (LLM) and vision language models (VLM) to the task of information extraction from invoice documents with diverse layouts, without relying on human-annotated data. Doculite includes (a) InvoicePy, an LLM driven synthetic invoice generator in text domain for training LLMs for the task of information extraction from invoice documents which are processed via optical character recognition (OCR) models, and (b) TemplatePy, an HTML-based synthetic invoice template generator in the image domain for training VLMs for information extraction from invoice document images. We also curate "Challenging Invoice Extraction dataset" containing 184 real world invoices[1]. The research is in collaboration with a Fintech startup that identifies itself as an "Agentic AI Platform for Finance and Accounting". Domain experts at the Fintech startup annotate the "Challenging Invoice Extraction dataset" and continuously evaluate the performance of LLM and VLM models trained using DocuLite. Experiments demonstrate that openchat-3.5-1210-7B LLM model trained with InvoicePy generated dataset achieves a 0.525 points improvement in the F1 score over the openchat-3.5-1210-7B LLM model trained with publicly available UCSF dataset on the "Challenging Invoice Extraction dataset". We also show that InternVL-2-8B VLM model trained with Templatepy generated dataset achieves a 0.513 points improvement in the F1 score over the InternVL-2-8B VLM model trained with publicly available UCSF dataset on the "Challenging Invoice Extraction dataset". To the best of our knowledge, Doculite is the first scalable and privacy preserving framework for adapting LLMs and VLMs for information extraction from invoice documents with diverse layouts.

Introduction

Businesses routinely process a large volume of visually rich and composite financial documents (VRDs), such as invoices, receipts, and purchase orders (Fashina 2024). Small businesses often handle hundreds of VRDs daily, with manual data extraction taking up to 10 to 15 minutes per document per staff member (Chetule 2024). For larger enterprises, extraction time scales significantly, as they process orders of magnitude more VRDs. To address this challenge, many organizations rely on dedicated finance document understanding models that automatically extract information from VRDs into structured formats (e.g., JSON or XML), which can be directly integrated into accounting software such as QuickBooks (Maheshwari and McLain 2006) (Šimsa et al. 2023). A central challenge faced by enterprises developing financial document understanding models is training them to standardise information extraction by accurately mapping extracted values to predefined field names, irrespective of layout variations. In addition, enterprises must support the rapid on-boarding of finance document understanding models for new businesses, a critical factor in reducing operational costs and maintaining a competitive edge (Company 2022).

Traditional rule-based and template-based methods (Baviskar et al. 2021) struggle with the layout diversity and mixed structured/unstructured content of financial documents (Musumeci et al. 2024). To address these limitations, enterprises increasingly adopt LLMs (input: <query, OCR text>) and VLMs (input: <query, document image>) for tasks like information extraction, question answering, and sentiment analysis (Dubey et al. 2024; Chen et al. 2024; Tanaka et al. 2024). However, deploying these models in the finance industry presents two major challenges: privacy constraints and task complexity. Privacy constraints limit access to real world training data, as institutions cannot share sensitive documents with commercial LLMs (e.g., GPT-4o (Hurst et al. 2024), Gemini (Team et al. 2023), Claude (Caruccio et al. 2024)) due to external hosting. Public datasets are not representative of real-world documents due to their limited structural complexity (Šimsa et al. 2023), and while synthetic data can help, generating realistic, domain-aligned samples is costly and time-intensive. Open-source models like LLaMA-3.3-70B (Dubey et al. 2024) and DeepSeek-r1 (Liu et al. 2024) provide an alternative but demand heavy compute, typically multiple A100s (Kwon et al. 2023), which many enterprises lack. Task-specific challenges arise from the high variability in VRD layouts (Xu et al. 2020), where key fields appear inconsistently, in close proximity, or within colliding structures (e.g., tables nested within tables), often requiring inference. Semantic ambiguities (e.g.,

[1]Due to privacy constraints, the dataset and code cannot be publicly released. However, a de-identified version of the Challenging Invoice Extraction dataset will be made available.

"freight charges" vs. "shipping charges", "receiver name" vs. "receiving person name") further complicate the extraction process (Douzon et al. 2023; Van Meerten et al. 2020). Addressing these issues requires privacy-preserving methods that enable smaller models (7B–8B) to match the performance of larger models (70B–405B) while maintaining high accuracy, efficiency, and adaptability in real world financial settings (VMware 2024). Our contributions are:

- Invoicepy, an LLM-driven framework to generate synthetic invoices in the text domain for the task of information extraction from invoice documents which are processed via OCR models. Experiments demonstrate that the openchat-3.5-1210-7B LLM model trained with Invoicepy generated dataset achieves a 0.525 points improvement in F1 score averaged across five "line-item fields"[2] on the "Challenging Invoice Extraction dataset" over the openchat-3.5-1210-7B LLM model trained on UCSF public dataset. To the best of our knowledge, this represents the first application of LLMs for generating synthetic invoices tailored to the complexities of real world invoice documents.
- Templatepy, an HTML-based synthetic invoice template generator in the image domain for training VLMs for information extraction from invoice document images. Experiments demonstrate that the Internvl-2-8B model trained with Templatepy generated dataset achieves a 0.513 points improvement in F1 Score averaged across five "line-item fields"[2] on the "Challenging Invoice Extraction dataset" over the Internvl-2-8B model trained with on UCSF public dataset. To the best of our knowledge, this is the first application of synthetic templates to train VLMs for accurate semi-structured information extraction from invoices, achieving performance on par with real-world data trained models.

Related Work

Document understanding has progressed rapidly with the emergence of multimodal transformers, layout-aware encoders, and OCR-free generative models. Early approaches relied on OCR and token classification over visually parsed layouts, but recent advances include end-to-end generation, vision-language pretraining, and instruction tuning. Donut (Kim et al. 2022) pioneered an OCR-free method for extracting structured data directly from document images, performing well on simple layouts but struggling with complex tables. Layout-aware models such as LayoutLM (Xu et al. 2020) and UDOP (Tang et al. 2023) remain effective for form-like documents, leveraging 2D positional embeddings and visual-text alignment. LayoutLMv3 (Huang et al. 2022) strikes a strong balance between performance and inference efficiency in enterprise settings. Specialized generative models like DocLLM (Wang et al. 2023) and DocOwl (Hu et al. 2024) adopt large-scale instruction tuning and enhanced layout encoding. DocLLM, trained on 11M real documents, achieves 82.8% Average Normalized Levenshtein Distance (ANLS) (Peer et al. 2024) at the 7B scale on DocVQA (Mathew et al. 2021). DocOwl introduces visual compression for efficient multi-page inference. General purpose LLMs (e.g., GPT-4 (Hurst et al. 2024)) using <OCR text> inputs achieve competitive zero-shot performance (82.8% ANLS), while multimodal models like InternVL 2.0 (Chen et al. 2024) reach 91.6% ANLS without extensive domain-specific tuning. General purpose models generalize well across formats, aided by flexible prompting. Model performance in document extraction depends critically on pretraining corpus scale, diversity, and architecture. LayoutLM and UDOP are trained on large synthetic datasets such as IIT-CDIP (Lewis et al. 2006) and DocVQA, using OCR tokens and image-text pairs. Donut relies on paired document images and structured outputs (e.g., SynthDoG (Kim et al. 2022)), while DocLLM and DocOwl integrate instruction tuning and multitask learning across synthetic and real sources. General purpose LLMs, in OCR+LLM pipelines, are pretrained on massive web-scale corpora and refined with document-specific fine-tuning. Despite their generalization capabilities, these systems depend heavily on accurate OCR and prompt design. While large-scale pretraining helps, data diversity and domain realism are essential. Public benchmarks including FUNSD (Xu et al. 2022), CORD (Park et al. 2019), and DocVQA—lack the structural complexity of real-world financial documents, often featuring regular layouts and annotation inconsistencies. As a result, models perform well on benchmarks but degrade on noisy, real world invoices. This underscores the need for privacy-preserving, domain-aligned synthetic data that reflects real-world variability for training models.

Methodology

In this section, we describe the methods, Invoicepy and Templatepy, introduced as part of DocuLite.

Invoicepy

Invoicepy is a synthetic data generation framework for training LLMs on invoice information extraction. It takes `<OCR, Human Annotated Label>` as input, where 'OCR' is the OCR model output for a real customer invoice and generates `<Synthetic OCR, Synthetically Annotated Label>` samples, where the 'Synthetic OCR' simulates OCR model output while preserving the layout and content structure of the customer invoice. Serving as a privacy-preserving anonymization layer, Invoicepy generates obfuscated but structurally faithful data, enabling domain-aligned model training without exposing sensitive information. The framework has two stages (Refer to Fig 1 in the Appendix): (1) **Generation** and (2) **Extraction**. In the generation stage, LLaMA-3-70B is instructed in a zero-shot Chain of Thought setup to generate a 'Synthetic OCR' from real customer invoice 'OCR' by replacing subwords, words, and numbers while preserving the original structure. A rationale is generated to guide output quality, and an optional second pass adds variability. In the extraction stage, Mixtral-8x22B-Instruct-v0.1 takes `<OCR, Human`

[2]Line items refer to fields detailing transaction components—product code, description, quantity, unit of measurement, unit price, and net amount—while entities encompass remaining fields, including names, addresses, IDs, and others.

Field	M1 (Few-shot)	M2 (UCSF)	M3 (Invoicepy)	M4 (Templatepy)
Product Code	0.31	0.35	**0.95**	0.93
Description	0.51	0.58	0.89	**0.95**
Quantity	0.20	0.23	**0.96**	0.92
Unit of Measurement	0.45	0.51	**0.94**	**0.94**
Net Amount	0.41	0.45	**0.97**	0.95
Unit Price	0.31	0.42	**0.98**	0.93

Table 1: F1 scores for line-item field extraction from invoices using different LLM and VLM configurations. M1: OpenChat-3.5-1210 with few-shot UCSF samples; M2: OpenChat-3.5-1210 fine-tuned on UCSF; M3: OpenChat-3.5-1210 fine-tuned on Invoicepy; M4: InternVL-2-8B fine-tuned on Templatepy.

`Annotated Label, Synthetic OCR>` as input and returns `<Synthetic OCR, Synthetically Annotated Label>` (Refer to Fig 2 in the appendix), resulting in final training pairs `<Synthetic OCR, Synthetically Annotated Label>`. Model choices are based on extensive experiments to ensure layout fidelity, annotation quality, and prevention of data leakage. Note that we use LLaMA-3-70B for the first stage and Mixtral-8x22B for the second stage because LLaMA excels at high-fidelity generative rewriting needed to produce layout-preserving synthetic OCR, whereas Mixtral provides cost-efficient yet reasonably accurate structured data extraction.

Templatepy

Templatepy addresses the limitations of public invoice datasets—limited layout diversity, inconsistent labeling—and the inaccessibility of proprietary data by eliminating the need for real or synthetic invoices during VLM training. Instead, it uses structured `html`-templates for scalable, privacy-compliant information extraction. VLM-based invoice extraction comprises three stages: (1) Token Extraction, where tokens are identified from the document image; (2) Token Grouping, where related tokens are clustered; and (3) Key-Value Mapping, where token groups are mapped to predefined fields—requiring domain-specific reasoning. Templatepy shifts training to focus on this final stage using synthetic templates that abstract layout while retaining structural logic. Since VLMs handle extraction and grouping via pretraining, Templatepy targets domain adaptation through key-value alignment. These layout-agnostic templates enable generalizable learning across invoice types and structured document domains. Templatepy takes as input a set of field names, their aliases, and a maximum number of rows, and generates document images with tables containing a random number of rows and columns. Each column header is randomly assigned a field name or alias. For entities[2], the number of rows is fixed at 1, with annotations as `<field name, row 0 column` j`>`, where j is the column index of `field name`. For line items[2], the table includes up to the specified maximum number of rows, with annotations as `<field name, row` i `column` j`>`, where i is the index of ith line item and j the column header index of the field name of that line item. Tables may or may not include borders, and cells may or may not contain values, both randomized. The output is `<Synthetic Image, Synthetically Annotated Output>` (Fig 3).

Dataset

We introduce the "Challenging invoice extraction dataset" of 184 annotated documents, curated to reflect real-world complexities in document understanding. Annotation was performed by 4 commerce graduates from Bengaluru, India (average age 20), supervised by a finance expert with 10 years of experience. Documents were classified as "challenging" based on: (a) presence of multiple tabular structures, (b) occurrence of nested or overlapping patterns (e.g., tables embedded within other tables), and (c) requirement of multi-step reasoning to infer values absent in the document (e.g., calculating the net amount for a line item using the formula: net amount = unit price $\times$ quantity, when only partial values are explicitly provided). Each document underwent 5 rounds of annotation by different annotators, followed by quality assurance and domain-specific validation to ensure only genuinely complex documents were included. A similar procedure was used to curate 375 complex training invoices from RVL-CDIP dataset, a subset of the UCSF public dataset (Larson, Lim, and Leach 2023). Notably, the public dataset lacks the structural complexity typical of real-world invoices.

Experiment and results

We perform LoRA fine-tuning (Dettmers et al. 2023) of the OpenChat-3.5-1210-7B LLM for invoice information extraction on training set of 375 invoices from the public dataset, and 184 synthetic invoices generated via running Invoicepy on the Challenging Invoice Extraction dataset. We use a maximum context length of 2048 tokens, batch size 2, LoRA rank 32, and LoRA alpha 64, for 3 training epochs. Training beyond 3 epochs or increasing the LoRA rank led to a performance in terms of F1 score. The best-performing checkpoints are selected based on F1 score on the Challenging Invoice Extraction dataset. We also fine-tune the InternVL-2-8B VLM on 650 synthetic invoices generated via running Templatepy. We use a context length of 4096 tokens, batch size 1, LoRA rank 64, LoRA alpha 128, and one training epoch. Additional training degrades performance. Best checkpoints are selected based on F1 score on the Challenging Invoice Extraction dataset.

We evaluate 4 model configurations for line-item[2] field extraction task from invoices using F1 score across six

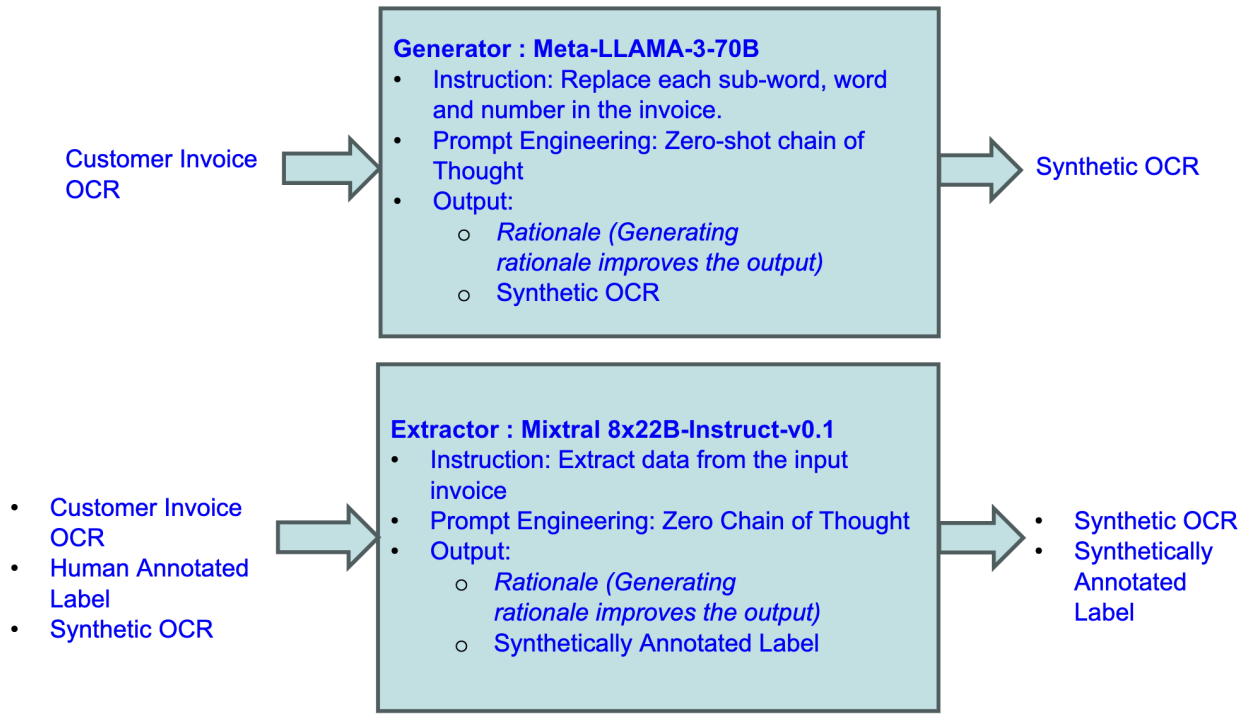

Figure 1: Invoicepy workflow diagram. The generation stage uses Meta-LLAMA-3-70B to transform real invoice OCR into structurally aligned synthetic OCR with rationale, preserving layout while obfuscating content. The extraction stage uses Mixtral-8x22B-Instruct-v0.1 to generate synthetically annotated labels from the synthetic OCR, guided by real OCR and human annotations. This two-step process produces high-quality, privacy-preserving training data for LLMs.

fields: *Quantity*, *Unit Price*, *Net Amount*, *Product Code*, *Description*. The models are: M1 (OpenChat-3.5-1210 with few-shot samples from the public dataset), M2 (OpenChat-3.5-1210 fine-tuned on the public dataset), M3 (OpenChat-3.5-1210 fine-tuned on the synthetic Invoicepy dataset), and M4 (InternVL-2-8B fine-tuned on the synthetic Templatepy dataset). Models trained on synthetic data (M3 and M4) consistently outperform the baselines (M1 and M2). M3 achieves the highest overall performance, surpassing M1 and M2 by 113% in average F1 score (Table 1), with M4 close behind at 111%. On the *Quantity* field, M3 improves by 76 and 73 points over M1 and M2, respectively; M4 also performs strongly with an F1 score of 0.92. For *Unit Price*, M3 outperforms M1 by 67 points, and M4 achieves 0.93. In *Net Amount*, M3 scores 0.97—52 points higher than M2—while M4 reaches 0.95. For *Product Code*, M3 leads with a 60-point advantage over M2, and M4 scores 0.93. In *Description*, M4 achieves the highest F1 score of 0.95, outperforming M2 by 37 points. Overall, M3 (Invoicepy) leads in 5 of 6 fields, demonstrating the effectiveness of synthetic datasets.

Summary, Conclusion, and Future Work

This work introduces Doculite, a scalable, privacy-preserving framework for adapting LLMs and VLMs to information extraction from VRDs. Doculite combines two synthetic data generation methods: InvoicePy for text-based synthetic invoices, and TemplatePy for template-based document images. Experiments on the Challenging Invoice Extraction dataset show that OpenChat-3.5-1210 trained on InvoicePy data achieves a 0.525 F1 improvement, while InternVL-2-8B trained on TemplatePy gains 0.513—both outperforming public-trained baselines. These results demonstrate synthetic data's effectiveness in improving extraction accuracy, reducing reliance on manual annotations, and enabling privacy-conscious training. Future work will extend support to multilingual invoices.

Limitations

The primary limitation of this study is its reliance on synthetic data, which, despite mimicking real-world structures, may not fully capture the variability and noise present in authentic invoices. Generalization beyond the current distribution therefore remains uncertain and requires more evaluation. Another notable limitation is the dependency on OCR quality in the LLM pipeline, which introduces susceptibility to extraction errors not fully reflected in the synthetic inputs. Additionally, the Challenging Invoice Extraction dataset, though structurally complex, offers limited domain diversity, representing a limited range of enterprise invoice formats. Finally, our evaluation is restricted to English documents; broader multilingual and non-Latin assessments are needed for wider applicability.

References

Baviskar, D.; Ahirrao, S.; Potdar, V.; and Kotecha, K. 2021. Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. *IEEE Access*, 9: 72894–72936.

Caruccio, L.; Cirillo, S.; Polese, G.; Solimando, G.; Sundaramurthy, S.; and Tortora, G. 2024. Claude 2.0 large language model: Tackling a real-world classification problem with a new iterative prompt engineering approach. *Intelligent Systems with Applications*, 21: 200336.

Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Chetule, S. 2024. Invoice Data Extraction: 3 Methods to Extract Invoice Data. Accessed: 2025-05-19.

Company, M. . 2022. Winning corporate clients with great onboarding. Accessed: 2025-05-19.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36: 10088–10115.

Douzon, T.; Duffner, S.; Garcia, C.; and Espinas, J. 2023. Improving Information Extraction on Business Documents with Specific Pre-Training Tasks. In *arXiv preprint arXiv:2309.05429*.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fashina, A. 2024. Transformation of Finance Communication with Vendors Using AI. Interview by Emily, Digital Transformation Consultant at Hyperbots.

Hu, A.; Xu, H.; Zhang, L.; Ye, J.; Yan, M.; Zhang, J.; Jin, Q.; Huang, F.; and Zhou, J. 2024. mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. *arXiv preprint arXiv:2409.03420*.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proceedings of the 30th ACM International Conference on Multimedia (ACM MM)*, 1190–1198.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Kim, G.; Hong, T.; Yim, M.; Nam, J.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; and Park, S. 2022. OCR-free Document Understanding Transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 13688, 485–503. Springer.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J.; Zhang, H.; and Stoica, I. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 611–626.

Larson, S.; Lim, G.; and Leach, K. 2023. On evaluation of document classification with rvl-cdip. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2665–2678.

Lewis, D.; Agam, G.; Argamon, S.; Frieder, O.; Grossman, D.; and Heard, J. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 665–666.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Maheshwari, S. K.; and McLain, M. P. 2006. Selection of Accounting Software Tools for Small Businesses: Analytical Hierarchy Process Approach. In *Allied Academies International Conference. Academy of Accounting and Financial Studies. Proceedings*, volume 11, 39. Jordan Whitney Enterprises, Inc.

Mathew, M. M.; Islam, M. R.; Shafait, F.; and Dengel, A. 2021. DocVQA: A Dataset for VQA on Document Images. 2200–2208.

Musumeci, E.; Brienza, M.; Suriani, V.; Nardi, D.; and Bloisi, D. D. 2024. Llm based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. In *International Conference on Human-Computer Interaction*, 98–117. Springer.

Park, S.; Shin, S.; Lee, B.; Lee, J.; Surh, J.; Seo, M.; and Lee, H. 2019. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *NeurIPS 2019 Workshop on Document Intelligence*.

Peer, D.; Schöpf, P.; Nebendahl, V.; Rietzler, A.; and Stabinger, S. 2024. ANLS*–A Universal Document Processing Metric for Generative Large Language Models. *arXiv preprint arXiv:2402.03848*.

Šimsa, Š.; Šulc, M.; Uřičář, M.; Patel, Y.; Hamdi, A.; Kocián, M.; Skalickỳ, M.; Matas, J.; Doucet, A.; Coustaty, M.; et al. 2023. Docile benchmark for document information localization and extraction. In *International Conference on Document Analysis and Recognition*, 147–166. Springer.

Tanaka, R.; Iki, T.; Nishida, K.; Saito, K.; and Suzuki, J. 2024. Instructdoc: A dataset for zero-shot generalization of visual document understanding with instructions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 19071–19079.

Tang, Z.; Yang, Z.; Wang, G.; Fang, Y.; Liu, Y.; Zhu, C.; Zeng, M.; Zhang, C.; and Bansal, M. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 19254–19264.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Van Meerten, J.; Wadden, D.; van Noord, R.; van Erp, M.; and van den Bosch, A. 2020. Extracting Fine-Grained Economic Events from Business News. In *Proceedings of the*

First Workshop on Financial Technology and Natural Language Processing, 236–246.

VMware. 2024. LLM Inference Sizing and Performance Guidance. Accessed: 2025-05-19.

Wang, D.; Raman, N.; Sibue, M.; Ma, Z.; Babkin, P.; Kaur, S.; Pei, Y.; Nourbakhsh, A.; and Liu, X. 2023. DocLLM: A layout-aware generative language model for multimodal document understanding. *arXiv preprint arXiv:2401.00908*.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 1192–1200. ACM.

Xu, Y.; Lv, T.; Cui, L.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; and Wei, F. 2022. XFUND: A Benchmark Dataset for Multilingual Visually Rich Form Understanding. 249–257.

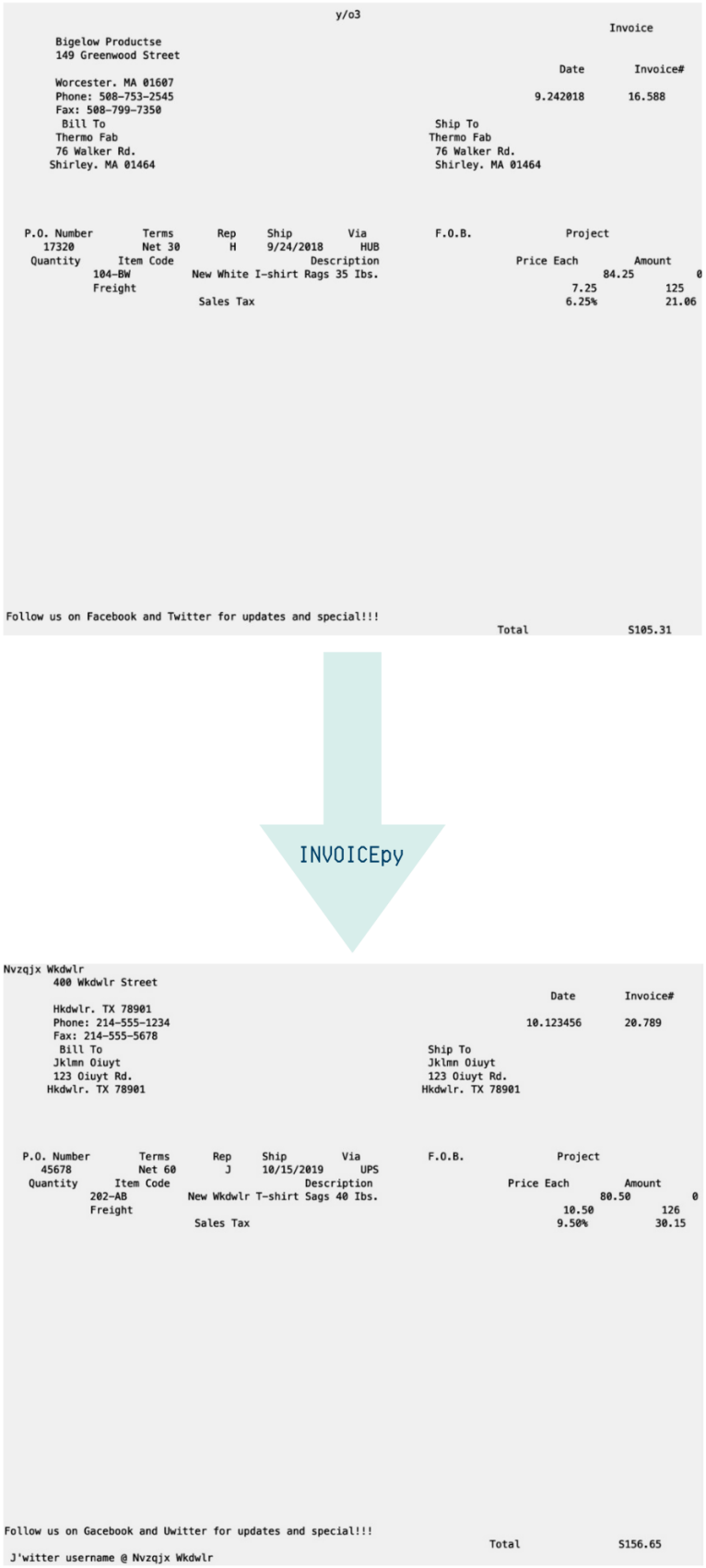

Figure 2: Invoicepy synthetic data generation process. The top shows a real invoice OCR model output, and the bottom shows a structurally aligned, anonymized 'Synthetic OCR' generated using LLaMA-3-70B, preserving layout while obfuscating content. This enables privacy-preserving training of LLMs for invoice information extraction.

d	Price Extension	Rate	Ship Quantity	UM	seq#	Item	Product Code	Amount	PO Number
Header Row 1, Column 1									
Row 1, Column 1	Row 1, Column 2		Row 1, Column 4		Row 1, Column 6			Row 1, Column 9	Row 1, Column 10
Header Row 2, Column 1									
			Row 2, Column 4			Row 2, Column 7	Row 2, Column 8		Row 2, Column 10
Header Row 3, Column 1									
Row 3, Column 1		Row 3, Column 3			Row 3, Column 6	Row 3, Column 7			Row 3, Column 10

Figure 3: Sample document image generated by Templatepy containing a synthetic table. Column headers are randomly assigned field names or aliases, and cells are filled with placeholder text representing structured information. This layout-agnostic design preserves structural logic while obfuscating content, enabling privacy-preserving, domain-aligned training of VLMs without reliance on real invoices.