Information Extraction from Visually Rich Documents using LLM-based Organization of Documents into Independent Textual Segments

Aniket Bhattacharyya[1], Anurag Tripathi, Ujjal Das, Archan Karmakar, Amit Pathak, Maneesh Gupta
Amazon
[1]aniketttb@amazon.com

Abstract

Information extraction (IE) from Visually Rich Documents (VRDs) containing layout features along with text is a critical and well-studied task. Specialized non-LLM NLP-based solutions typically involve training models using both textual and geometric information to label sequences/tokens as named entities or answers to specific questions. However, these approaches lack reasoning, are not able to infer values not explicitly present in documents, and do not generalize well to new formats. Generative LLM-based approaches proposed recently are capable of reasoning, but struggle to comprehend clues from document layout especially in previously unseen document formats, and do not show competitive performance in heterogeneous VRD benchmark datasets. In this paper, we propose BLOCKIE, a novel LLM-based approach that organizes VRDs into localized, reusable semantic textual segments called *semantic blocks*, which are processed independently. Through focused and more generalizable reasoning,our approach outperforms the state-of-the-art on public VRD benchmarks by 1-3% in F1 scores, is resilient to document formats previously not encountered and shows abilities to correctly extract information not explicitly present in documents.

1 Introduction

Visually Rich Document Understanding (VRDU) is a well researched topic due to its wide industry applicability. Structured or semi-structured documents such as invoices, forms, contracts, receipts etc are handled by most organizations, and for large organizations the volume of such documents can be massive. Processing these documents, especially those of a financial or legal nature, is vital. Figure 1 shows a typical application of VRDU. As can be seen, an ideal information extraction or processing solution, should have the following desiderata -

- High-quality extraction - High precision and recall of desired entities (such as company name or address) to be extracted.
- Handling heterogeneity of formats and languages - Handling documents from various sources with different templates (legal fax from US and supplies store invoice from Indonesia in Figure 1). Public datasets such as Lewis et al., 2006 illustrate the degree of heterogeneity found in real life applications.
- Handling new document formats - Solution should be able to handle documents with formats not seen during its training to avoid failure in production environment.
- Ability to perform value-absent inference - Entities to be extracted (such as number of line items in Figure 1) may not always be present explicitly, and may need to be inferred.

A typical approach to document information extraction begins with Optical Character Recognition (OCR) using tools like Amazon Textract or Tesseract (Hegghammer, 2022). However, OCR alone fails to address several key challenges. Documents exhibit diverse formats and structures, requiring spatial reasoning to correctly associate text with their semantic roles. Systems must understand contextual relationships - for instance, recognizing that 'CGST', 'VAT', and 'SR' all represent tax types, or identifying a vendor name without explicit labels. Additionally, solutions must generalize across heterogeneous document layouts and languages.

Recent approaches have attempted to address these challenges through layout-aware NLP models (Xu et al., 2020; Huang et al., 2022; Peng et al., 2022; Luo et al., 2023) enhance text processing with spatial information through mechanisms using cross-attention between text and bounding box embeddings. While effective for template-matching, we show that these models struggle with generalizing to new document formats, making inferences

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 17241–17256
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

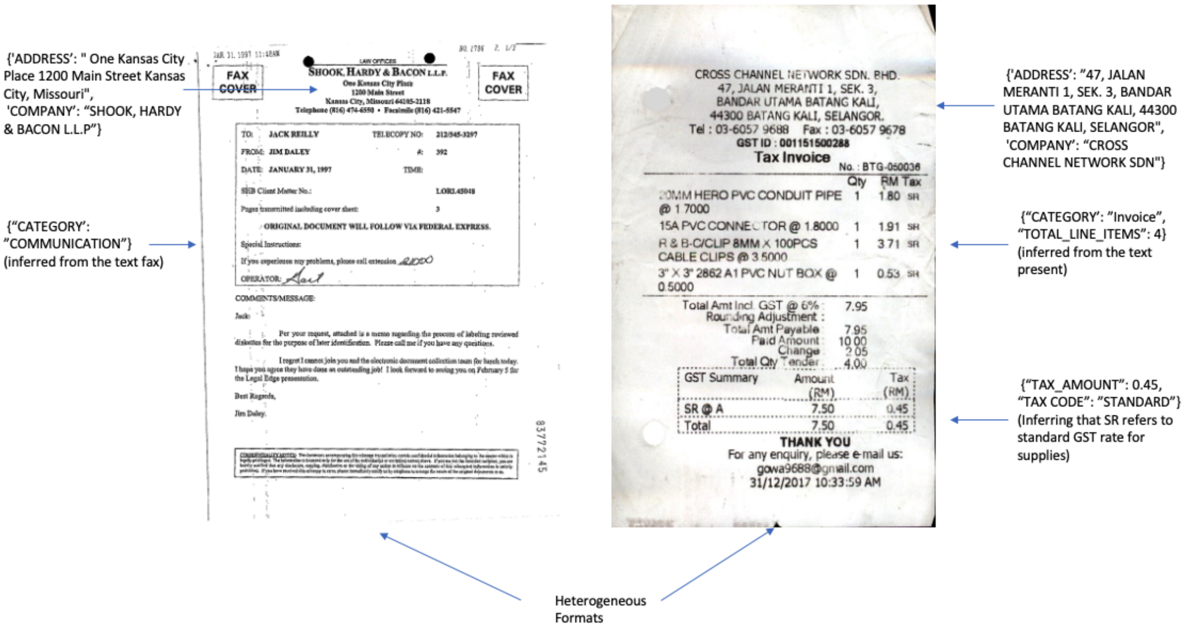

Figure 1: The Information Extraction Task, illustrated using sample images from (Jaume et al., 2019) and (Huang et al., 2019)

about implicit or absent values, and understanding semantic relationships beyond training examples.

Large Language Models have demonstrated strong reasoning capabilities through chain-of-thought demonstrations (Wei et al., 2023) and few-shot examples attached to the prompt (Brown et al., 2020). However, LLMs face their own limitations: they struggle with processing documents dissimilar to few-shot examples, handling complex layouts efficiently, and scaling prompts for multiple entity extraction. Even approaches using dynamic example selection based on document similarity (Perot et al., 2024) require at least one document with matching format in the labeled sample.

In this work, we propose BLOCKIE, a novel information extraction algorithm that leverages *semantic block*-level parsing. Our approach first identifies self-contained groups of text tokens (*semantic blocks*) and processes them using LLM-driven reasoning informed by similar blocks from labeled samples (see Figure 8 for an example on how documents with different templates can have similar blocks). Since semi-structured documents naturally organize information in human-readable blocks (Figure 6), this localized reasoning generalizes well across different document formats. BLOCKIE mimics human document processing by first understanding local regions (**B**lock **L**evel **O**rganization) and then leveraging **C**ontextual **K**nowledge from other blocks to stitch information together for IE.

We show that our approach outperforms the state-of-the-art on public benchmark datasets and satisfies all the desiderata for an IE solution. To summarize, we make the following contributions:

- We introduce BLOCKIE: Block-Level Organization and Contextual Knowledge-based Information Extraction, a novel algorithm for VRDU that organizes documents into self-contained segments of text tokens called semantic blocks, which are processed using reasoning that generalizes across document formats.

- We apply BLOCKIE to public benchmark datasets CORD, FUNSD and SROIE, and show that our method concurrently outperforms the current state-of-the-art on all these three datasets by 1-3% in F1 score.

- We show that block-level reasoning makes BLOCKIE robust to heterogeneous document databases and new document formats, prevents degradation of performance with smaller LLMs, and allows LLMs to perform value-absent inference.

2 Related Work

Prior work in VRD understanding can be broadly categorized into three approaches: traditional methods, layout-aware models, and large language models. We discuss each in turn, highlighting their capabilities and limitations.

Traditional Methods initially relied on rule-based systems and handcrafted features (O'Gorman, 1993; Ha et al., 1995; Simon et al., 1997; Marinai et al., 2005; Mausam et al., 2012; Chiticariu et al., 2013). While these approaches worked for known templates, they failed to generalize to new document formats. Later deep learning approaches leveraged RNNs (Aggarwal et al., 2020; Palm et al., 2017), CNNs (Hao et al., 2016; Denk and Reisswig, 2019; Katti et al., 2018), and transformers (Wang et al., 2023c; Majumder et al., 2020) to extract structural information from documents. However, these methods required extensive component-level labeling, limiting their practical applicability.

Layout-aware NLP Models enhanced traditional approaches by incorporating document layout information. Several architectural innovations were proposed: Powalski et al. (2021) introduced the usage of generative transformers for document understanding. This was followed by works such as Appalaraju et al. (2021); Hwang et al. (2021); Bai et al. (2022); Dhouib et al. (2023). Other proposed approaches include layout-aware language models combining BERT-style architectures (Devlin et al., 2019; Liu et al., 2019; Bao et al., 2020) with spatial information through learnable modules, 2D position embeddings (Xu et al., 2020), and attention mechanisms (Xu et al., 2022; Huang et al., 2022; Peng et al., 2022). Further advances introduced geometric pre-training (Luo et al., 2023), graph contrastive learning (Lee et al., 2023), and unified frameworks for simultaneous text detection and classification (Yang et al., 2023). Recent work has improved these models through reading-order prediction (Zhang et al., 2024). While these approaches achieve strong performance when fine-tuned on benchmark datasets like DocVQA (Mathew et al., 2021) and FUNSD (Jaume et al., 2019) after pre-training on large document corpora like IIT-CDIP (Lewis et al., 2006), they remain limited by their token-classification approach, requiring explicit answer presence and struggling with new document formats.

Large Language Models represent the newest approach to VRD understanding. Commercial models like Claude (Anthropic, 2024c) and ChatGPT (OpenAI, 2023) demonstrate zero-shot reasoning capabilities, with Claude 3 achieving state-of-the-art performance on DocVQA (Anthropic, 2024b). Open-source models like LLaVa (Liu et al., 2023) and CogVLM (Wang et al., 2024) show promise on visual question answering tasks but struggle with zero-shot and multi-entity extraction (Bhattacharyya and Tripathi, 2024).

Recent work has explored specialized LLM applications for information extraction, particularly in Named Entity Recognition (Keraghel et al., 2024; Laskar et al., 2023; Ashok and Lipton, 2023; Wang et al., 2023b). For VRD-specific challenges, researchers have developed layout-aware pre-training (Luo et al., 2024), disentangled spatial attention (Wang et al., 2023a), and normalized line-level bounding box representations (Perot et al., 2024). However, these approaches have yet to surpass layout-aware NLP methods, and attempts to convert generative models to token-labeling systems often sacrifice their inference capabilities.

3 Semantic Blocks in VRDs

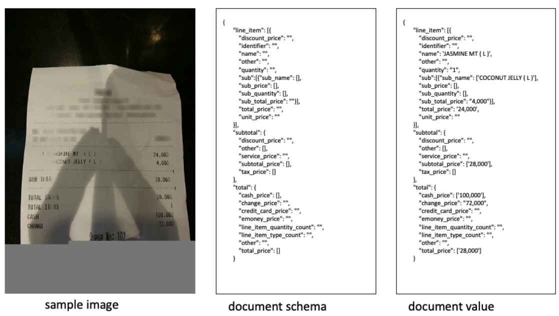

Figure 2: Sample image with document schema and value

In this section, we define the concept of semantic blocks theoretically, and we show how these are created practically in section 4.

Let us consider a set of documents $\mathcal{D}$ with a common set of hierarchical entities of interest $\mathbb{E}$, which we refer to as the document schema. Let $\mathcal{V}$ denote the set of all possible instantiations of $\mathbb{E}$. Given a document $D \in \mathcal{D}$, let $V_{\mathbb{E}}(D) \in \mathcal{V}$ denote the actual values of the entities $\mathbb{E}$ for D (for reference, consider sample document, schema and value in Figure 2).

For a document $D \in \mathcal{D}$, let $\mathcal{B}_{\mathcal{D}}$ denote the set of all possible segments (i.e. localized visual regions) of D. For any segment $B \in \mathcal{B}_{\mathcal{D}}$, let $V_{\mathbb{E}}(B)$ represent the document values with only entities present in B populated, other entities being blank. Note that $D \in \mathcal{B}_{\mathcal{D}}$ is a special segment comprising of the entire document.

The annotation operation can be thought of as an attempt to map a segment of a document to the document schema. As input, it takes in the target document segment, and parses it in the context of a larger segment with respect to the schema. The context segment could be any superset of the target, including (typically) the target segment itself or the entire document. Figure 3 illustrates the annotation operation with a target and context segment.

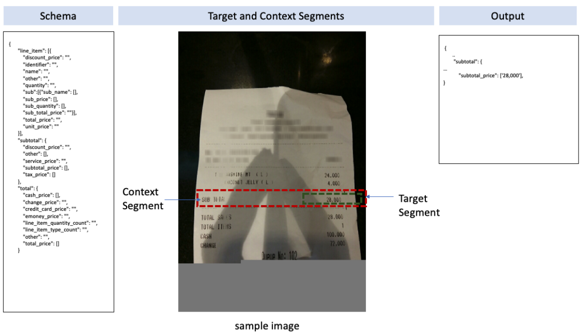

Figure 3: Sample image with document schema and value

Formally, for a given document schema $\mathbb{E}$, the annotation operation can be defined as a mapping $v : \mathcal{B}_{\mathcal{D}} \times \mathcal{B}_{\mathcal{D}} \mapsto \mathcal{V}$. If the annotation is correct, we have,

$$v(B, D) = V_{\mathbb{E}}(B), \forall B \in \mathbb{B}_{\mathbb{D}}, \forall D \in \mathbb{D} \quad (1)$$

Now, consider any segment $B \in \mathcal{B}_{\mathcal{D}}$ for a $D \in \mathcal{D}$. We define B as a *semantic block* if and only if:

$$v(B, B) = v(B, D) = V_{\mathbb{E}}(B) \quad (2)$$

In other words, a *semantic block* must be interpretable independently without any additional context - the values extracted from B in isolation must match those extracted with full document context.

To illustrate, consider Figure 2. In this example, B_1: (SUB TOTAL 28.000) is a semantic block with:

$$v(B_1, D) = subtotal : \{subtotal\_price : [28.000]\}$$

and B_2: (TOTAL SALE 28.0000) is a semantic block with:

$$v(B_2, D) = total : \{total\_price : [28.000]\}$$

On the other hand, (COCONUT JELLY (L), 4.000) cannot be a semantic block, as without the context of (1 JASMINE MT (L) 24.000), it is not possible to determine whether it is a sub-item and, if so, which line item it is a sub-item of.

Now, to create semantic blocks **in practice**, we introduce the concept of semantic *atoms* - the fundamental units for information extraction from VRDs. A semantic atom is an indivisible visual region containing text that forms a complete semantic unit while maintaining spatial coherence through proximity as well as horizontal or vertical alignment. The key characteristic of a semantic atom is that it cannot be decomposed further without losing its intended meaning. For example, in Figure 2, "TOTAL ITEMS" forms a semantic atom because splitting it into "TOTAL" and "ITEMS" individually would lose the specific meaning of 'number of items' - "TOTAL" alone could refer to price or quantity, while "ITEMS" alone loses specificity. Moreover, these words maintain spatial coherence through horizontal proximity in the document. Conversely, "TOTAL ITEMS 1", although coherent semantically and linked as an attribute value pair, is not spatially proximate, and hence is not an atom, but makes up two linked semantic atoms.

Note that there could be two different types of linkages between semantic atoms in a VRD - linkages of the form attribute:value, or linkages of hierarchy. By hierarchically linked semantic atoms we refer to semantic atoms that belong to hierarchical entities in the document schema. In practice, semantic blocks are *collections of semantic atoms*, such that *all linkages for each atom in the collection is present inside the collection itself*. This is a sufficient condition for equation 2, as given a schema, all context needed to parse any group of atoms is present in a collection of atoms linked to it as hierarchically or as attribute-value. To continue the example, (TOTAL SALE 28.0000) and (SUB TOTAL 28.000) are linked semantic atoms, and (1 JASMINE MT (L) 24.000 COCONUT JELLY (L), 4.000) are linked semantic atoms.

This theoretical foundation guides our development of practical algorithms for document processing, as we will demonstrate in subsequent sections. By decomposing documents into smaller, more generalizable semantic blocks, we can better handle the complexities of varying layouts while maintaining the semantic relationships crucial for accurate information extraction. In the following section, we show how BLOCKIE identifies and parses semantic blocks.

4 Proposed Methodology

Given a group of documents and a required set of entities that need to be extracted in the form of document schema, we first divide the document into a collection of semantic blocks of related text using LLMs. **In practice**, LLMs are used to identify semantic blocks. They are asked to break all of the text present in a document into blocks, where all related text should be present in the same block. Related text is defined in the prompt itself as text belonging to linked entities or hierarchical entities from the document schema, which is a sufficient condition for equation 2. The exact block creation template is provided in appendix A.

These blocks are then processed, which allows LLMs to develop generalizable abstract rules for IE. These partial block parses are then combined to return the set of entities required. However, prior to these steps, it is necessary to convert the train dataset labels to appropriate format, i.e. to independent blocks and their annotations, so that these can be used as few-shot examples during inference. Further details on each of these steps are provided below.

Train Dataset Labelling The train dataset is used as a labelled sample. VRD benchmarks such as Park et al., 2019 generally contain ground truth labels in a key-value format, with appropriate hierarchy and linkages. These are passed to an LLM along with document schema to return three things - (1) step-by-step reasoning for choosing a segment as a block (i.e. self-contained segments of linked atoms, as defined in section 3), (2) the words in the block, and (3) the partial annotation of the block, using the ground truth labels. All of these three outputs are used downstream. Appendix A contains the prompt used to extract these elements.

4.1 Block Creation

Given a document from the test dataset, we prompt the LLM to create blocks using the document schema, OCR text and bounding boxes, and dynamic few-shot examples from the labelled train dataset using cosine similarity of OCR text[1].The LLM leverages the step by step reasoning from the train dataset blocks on the few-shot samples to understand when a text segment can be considered a block. Note that while we used OCR text and bounding boxes, for multimodal LLMs one can pass the image directly. The creation of self-contained blocks is crucial; in section 5, we evaluate the impact of block creation on overall accuracy.

4.2 Block Parsing

Once blocks have been created, these are annotated by block parsers. As shown in figure 6, similar semantically meaningful blocks are found even in documents with different formats. Since these blocks are self-contained, they can be parsed independently.

The document schema is passed to the LLM with few-shot examples of the most similar blocks. The step-by-step reasoning of train dataset block parser triggers similar reasoning in the block parser, and the document schema guides it to return structured output in required format.

Figure 7 shows how the same example with similar blocks would be annotated by the block parser.

4.3 Combining Blocks

Finally, the document schema, blocks and their parses are provided to LLMs to return the entire filled out schema. The LLM acts as a judge assessing the block-parsing reason from the previous steps to stitch together the filled out document schema. Each semantic block benefits from being compared with similar blocks in other documents (which may be heterogenous), and the document schema guides the llm to return structured output.

Figure 4 illustrates these three steps using a sample document and schema.

Prompting Strategy We designed prompts for block creation, block parsing and block combining with Claude 3.5 Sonnet. We did not separately tune prompts for other LLMs as we wanted to test both BLOCKIE's generalizability as well as the lift that is obtained purely due to the design of BLOCKIE, rather than prompt tuning. Detailed prompts for all the stages are provided in the appendix.

5 Experimental Setup and Results

We designed our experimental evaluation to rigorously assess BLOCKIE's effectiveness in addressing these challenges. Our analysis examines the

[1]Perot et al., 2024 show that using similar documents in in-context learning examples improves performance in VRDs.

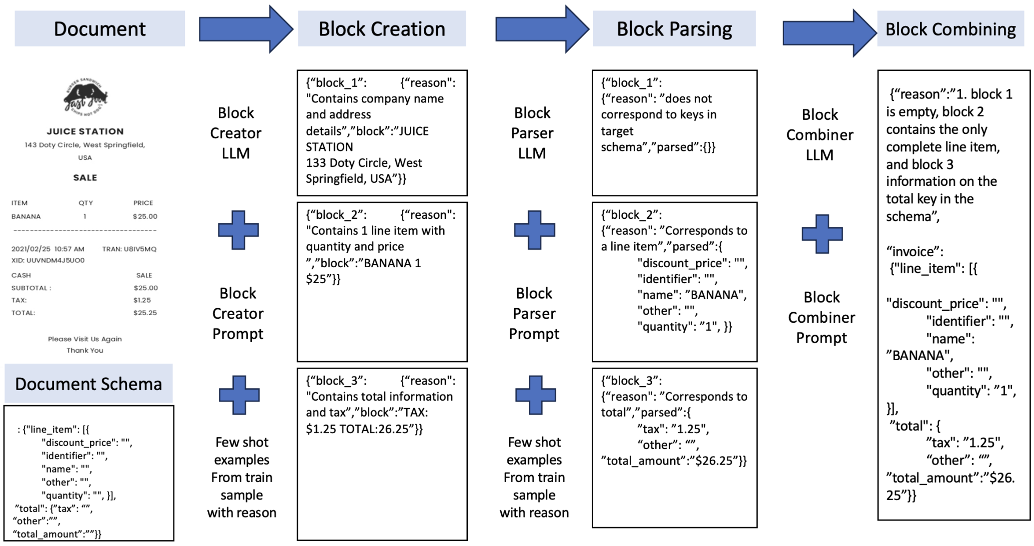

Figure 4: Illustrative flow with a simulated receipt and schema resembling CORD output requirement. The schema is passed along with the output of the block creator along with parses of similar blocks to block parser. Parsed blocks with target schema are then passed to get final output. Reasons are output at each stage.

method stands up against the desiderata for an ideal information extraction solution for a large heterogenous document database.

5.1 Experimental Setup

We evaluate BLOCKIE on three established information extraction benchmarks: CORD (Park et al., 2019), which focuses on restaurant receipts with hierarchical field structures; FUNSD (Jaume et al., 2019), a subset of Harley et al.; and SROIE (Huang et al., 2019), a receipt information extraction dataset. For FUNSD, we focus on entity linking as the original semantic entity classifications (question, answer, header, others) are not meaningful and do not align with real-world information extraction requirements.

To assess the generality of our approach, we conduct experiments for BLOCKIE with multiple language models of varying parameter counts: Claude 3.5 Sonnet (Anthropic, 2024a) and four variants of Qwen 2.5 (Qwen et al., 2025) with 7B, 14B, 32B, and 72B parameters respectively. We used 5 few shot-examples in the prompts for both block creator and parser. Following standard practice in document information extraction, we use the F1 score as our primary evaluation metric. For performance comparison, we consider state-of-the-art methods discussed in section 2, and we also conduct additional experiments with LayoutLMV3 (Huang et al., 2022) to show the limitations of layout-aware NLP methods. Additional details about the datasets and implementations are present in Appendix B.

5.2 Results

5.2.1 Performance Analysis

Table 1 presents BLOCKIE's performance compared to existing approaches across all three datasets. Using Sonnet as the base LLM, BLOCKIE achieves state-of-the-art performance, surpassing both traditional layout-aware approaches and recent LLM-based methods. Notably, BLOCKIE achieves 98.83% F1-score on CORD, 92.15% on FUNSD, and 98.52% on SROIE, establishing new benchmarks across all datasets. To verify that these improvements stem from our block-based methodology rather than just LLM capabilities, we compare against zero-shot and few-shot variants of Sonnet. The performance gap between BLOCKIE and these baseline approaches (shown in Table 1) demonstrates that the improvements arise from our semantic block methodology rather than raw LLM capabilities.

5.2.2 BLOCKIE helps smaller LLMs outperform large LLMs

We examine BLOCKIE's robustness to LLMs by evaluating performance across LLMs of varying

Approach	Method	FUNSD	CORD	SROIE
		EL	SER	SER
Layout-Aware NLP	DocTr(Feng et al., 2022)153M	73.9	98.2	-
	LayoutLMv3(Huang et al., 2022)368M	79.37	96.98	96.12
	DocFormer(Appalaraju et al., 2021) 502M	-	96.99	-
	FormNetLee et al. (2023) large	-	97.28	-
	ERNIE-Layout(Peng et al., 2022)large	-	97.21	97.55
	GeoLayoutLM(Luo et al., 2023)399M	88.06	98.11	96.62
	ESP(Yang et al., 2023)50M	88.88	95.65	-
	RORE-GeoLayoutLM (Zhang et al., 2024) 399M+24	88.46	98.52	96.97
LLM	DocLLM(Wang et al., 2023a)	-	67.4	91.9
	LMDX-Gemini Pro(Perot et al., 2024)	-	95.57	
	LayoutLLM(Luo et al., 2024)	-	63.1	72.72
	Sonnet - Zero shot	-	88.92	91.37
	Sonnet - Few shot	-	95.72	96.72
Ours	BLOCKIE - Sonnet	**92.15**	**98.83**	**98.52**

Table 1: Performance Comparison. BLOCKIE-Sonnet outperforms the state-of-the-art across all three datasets

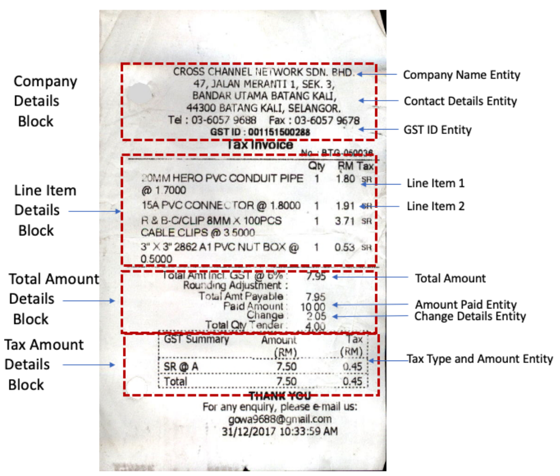

Figure 5: Motivating example for the conceptualization of VRD IE as the parsing of related semantic entities organized in blocks. The entities within a block are related which allows a human to understand that the address in the company details block belongs to the invoicing company instead of say the customer.

sizes. As shown in Table 2, BLOCKIE maintains strong performance even with smaller models - BLOCKIE with Qwen 2.5 32B (96.14% F1) outperforms LMDX-Gemini Pro (200B parameters, 95.57% F1) and Sonnet Zero-Shot as well as Few-shot (91.37% and 95.72% respectively), while BLOCKIE with Qwen 2.5 7B (87.72% F1) significantly surpasses other approaches using similar-sized models like DocLLM (67.4% F1) and LayoutLLM (63.1% F1). Note that the finetuned version of the Qwen 32B model falls short of Sonnet Few shot significantly (91.08% vs 95.72%), showing that the improvement in performance is caused by BLOCKIE and not purely the abilities of the LLM.

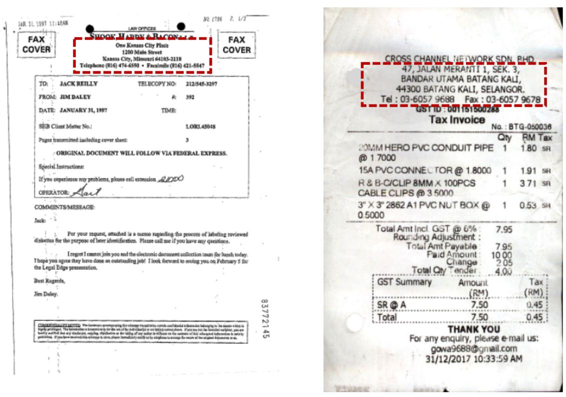

Figure 6: Two documents with different formats (a fax from a legal firm and a supplies store invoice) sharing a similar semantic block corresponding to contact information

5.2.3 BLOCKIE is resistant to heterogeneity and to unseen document formats.

To assess format resilience, we conduct two experiments. In the first experiment, we evaluate performance when training on only 100 samples selected for maximum format diversity (based on maximising text embedding distances with the test sample). Table 3 shows that while LayoutLMV3's performance drops significantly from 96.98% to 78.79% with diverse samples, BLOCKIE maintains robust performance (94.47% F1), demonstrating better generalization to format variations. This is even better that 91.48% achieved by Perot et al., 2024 by training on 100 random samples.

In our second experiment, we evaluate cross-

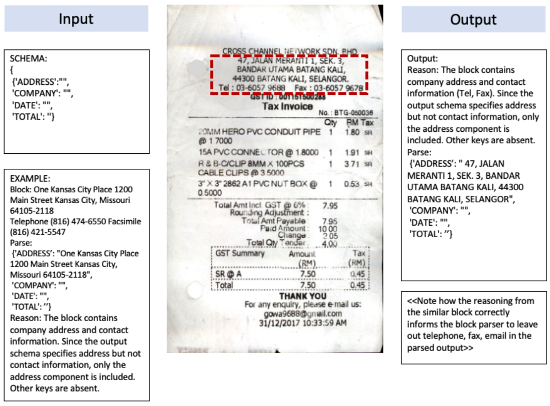

Figure 7: Block Parser on Figure 6, where the legal firm fax is used as a labelled train dataset example, and the supplies store invoice is treated as a test sample

Approach	CORD - SER
DocLLM - 7B	67.4
LayoutLLM - 7B	63.1
LMDX - Gemini Pro	95.57
Qwen 2.5 7B Finetuned	84.03
Qwen 2.5 14B Finetuned	89.36
Qwen 2.5 32B Finetuned	91.08
Sonnet - Zero shot	91.37
Sonnet - Few shot	95.72
BLOCKIE - Qwen 2.5 7B	87.72
BLOCKIE - Qwen 2.5 14B	89.98
BLOCKIE - Qwen 2.5 32B	**96.14**
BLOCKIE - Qwen 2.5 72B	**96.01**
BLOCKIE - Sonnet 3.5	**98.83**

Table 2: BLOCKIE with smaller LLMs outperforms massive state-of-the-art models Sonnet and Gemini Pro

dataset generalization by testing a CORD-trained model on SROIE documents (using the enity total amount, which is common in both datasets). As shown in Table 3, BLOCKIE maintains strong performance (97.06% F1) while LayoutLMV3's performance deteriorates substantially (33.43% F1), further validating our approach's resilience to format changes.

5.2.4 Block creation is crucial for BLOCKIE performance

The effectiveness of BLOCKIE relies critically on accurate semantic block creation. Our analysis reveals that block creation quality strongly correlates with final extraction performance (Table 4). The performance gap between different model sizes can be largely attributed to their block creation capabilities - Qwen 32B and 72B achieve state-of-the-art performance due to superior block creation (85.03% and 81.69% block-level F1[2] respectively), while smaller models show lower block creation accuracy.

To isolate the impact of block creation, we evaluate smaller models (7B, 14B) using ground truth blocks and blocks created by the 32B model. As shown in Table 5, with perfect blocks, even 7B and 14B models achieve performance comparable to larger models (94.38% and 94.98% F1 respectively), closing 80% of the performance gap, indicating that **block creation quality is the primary performance bottleneck**.

Interestingly, table 4 shows that the 32B model outperforms the 72B model in both block creation accuracy and overall F1 score. We also compared the capability of these two models to perform block parsing and combining. We conducted an experiment using the CORD dataset. We provided ground truth blocks (generated using Sonnet 3.5 with ground truth labels) and evaluated the performance of both the 32B and 72B models in parsing and combining these blocks. The results revealed that the 32B model achieved an F1 score of **98.13%**, while the 72B model scored **97.54%**. This suggests that, in our specific setup, the 32B model outperforms the 72B model in both block creation and subsequent parsing and combining tasks. However, overall, the block creation step remains the most crucial in determining performance.

5.2.5 BLOCKIE is able to perform value-absent inference

Finally, we demonstrate BLOCKIE's reasoning capabilities through value-absent inference. We evaluate on CORD receipts where line item counts are not explicitly stated but can be inferred through counting. On a sample of 20 such cases, BLOCKIE successfully infers the correct count in 18 instances (90% accuracy), handling complex scenarios including implicit quantities and hierarchical items. Figure 5.2.5 illustrates several challenging cases where BLOCKIE successfully performs multi-step reasoning to arrive at correct inferences. This capability distinguishes BLOCKIE from existing approaches that are limited to extracting explicitly present information.

[2]Block level F1 is derived by comparison with ground truth blocks created using labelled data

Test on	CORD - SER	SROIE -Total Amount
Trained on	[100 train samples least similar to test]	[train samples from CORD]
LayoutLMV3	78.79	33.43
Sonnet 3.5 Few Shot	92.11	95.39
BLOCKIE - Qwen 2.5 32B	86.51	91.01
BLOCKIE - Sonnet 3.5	**94.47**	**97.06**

Table 3: Resilience to heterogeneity and new formats. Sonnet is more resilient than LayoutLMV3, and BLOCKIE further enhances this resilience, outperforming layout-aware NLP methods designed to recognize templates.

Approach	CORD - SER Block F1	Entity F1
BLOCKIE - Qwen 2.5 7B	74.91	87.72
BLOCKIE - Qwen 2.5 14B	73.25	89.98
BLOCKIE - Qwen 2.5 32B	85.03	96.14
BLOCKIE - Qwen 2.5 72B	81.69	96.01
BLOCKIE - Sonnet 3.5	86.73	98.83

Table 4: Correlation between block creation accuracy and performance.

BLOCKIE Qwen Size	End To End	Qwen 32B blocks	Ground Truth Blocks
7B	87.72	90.91	94.38
14B	89.98	92.23	94.98

Table 5: Semantic Block F1-scores. After correcting semantic blocks of test samples, smaller models are able to recover 80% of the 10 percent performance gap with larger models

Figure 8: Some challenging inferences made by BLOCKIE. In test_30, the single line item does not have a quantity mentioned. In test_29, the LLM has to reason to leave out sub-items from the count. In test_20, it has to perform a multi-step addition.

6 Conclusion

In this work, we introduced the concept of **semantic blocks** and proposed a novel LLM-based approach for information extraction from documents leveraging them. The segmentation of documents into generalizable, smaller, self-contained semantic blocks allowed LLMs to generate focused step-by-step reasoning guiding their annotation, and we demonstrated that this was effective by showing state-of-the-art performance across diverse public datasets.

The framework is designed to be generalizable across various large language models (LLMs) and resilient to unseen document layouts and formats, and we demonstrated robust performance across multiple LLMs, heterogeneity and new, unseen document formats. Additionally, we also showcased the ability of BLOCKIE to perform value-absent inference.

The combination of semantic reasoning, robust generalization, and resilience to variation positions this methodology as a promising direction for future research in document information extraction. Future work could focus on incorporating image-based features such as font size, qualities such as bold/italics, etc, into semantic block creation even in text-only LLMs.

Limitations

We acknowledge the limitations of BLOCKIE with a view to motivating further research in this field. The computational architecture currently requires sequential LLM calls for block creation, processing and combining which increases latency. While our block creation methodology showed robust performance across all three datasets and experiments, it could be refined further. Specifically, the current block creation methodology does not leverage image-based contextual clues such as font, italics/bold, visual markers for linkages such as arrows, etc. Additionally, while robust performance was observed across 5 different LLMs of varying sizes, BLOCKIE's performance is inherently tied to the reasoning capability of the LLM being used. As was shown in section 5.2.4, it is vital to ensure that the LLM is able to reason and create proper blocks with linked semantic atoms, as missed linkages can be hard to recover. Future research should focus on robust block creation using the definition of semantic blocks and linked semantic atoms. Most of the testing focused on single page invoice-like documents. While it was shown that it is possible to bridge the performance gap between LLMs and specialized methods such as LayoutLMV3 on these documents (and even outperform these), more testing needs to be done on multi-page documents, complex elements like tables, figures etc within documents, and general VQA benchmarks to assess BLOCKIE's applicability to broader VQA tasks. Finally, using proprietary LLMs like Sonnet can make BLOCKIE less transparent even with step-by-step reasoning output, and caution needs to be exercised to ensure outputs are as expected.

References

Milan Aggarwal, Hiresh Gupta, Mausoom Sarkar, and Balaji Krishnamurthy. 2020. Form2Seq : A framework for higher-order form structure extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3830–3840, Online. Association for Computational Linguistics.

Anthropic. 2024a. claude-3-5-sonnet.

Anthropic. 2024b. The claude 3 model family: Opus, sonnet, haiku.

Anthropic. 2024c. Introducing the next generation of claude.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. Docformer: End-to-end transformer for document understanding. *Preprint*, arXiv:2106.11539.

Dhananjay Ashok and Zachary C. Lipton. 2023. Promptner: Prompting for named entity recognition. *Preprint*, arXiv:2305.15444.

Haoli Bai, Zhiguang Liu, Xiaojun Meng, Wentao Li, Shuang Liu, Nian Xie, Rongfu Zheng, Liangwei Wang, Lu Hou, Jiansheng Wei, Xin Jiang, and Qun Liu. 2022. Wukong-reader: Multi-modal pre-training for fine-grained visual document understanding. *Preprint*, arXiv:2212.09621.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. *Preprint*, arXiv:2002.12804.

Aniket Bhattacharyya and Anurag Tripathi. 2024. Information extraction from heterogeneous documents without ground truth labels using synthetic label generation and knowledge distillation. *Preprint*, arXiv:2411.14957.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 827–832, Seattle, Washington, USA. Association for Computational Linguistics.

Timo I. Denk and Christian Reisswig. 2019. Bertgrid: Contextualized embedding for 2d document representation and understanding. *Preprint*, arXiv:1909.04948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Mohamed Dhouib, Ghassen Bettaieb, and Aymen Shabou. 2023. Docparser: End-to-end ocr-free information extraction from visually rich documents. *Preprint*, arXiv:2304.12484.

Hao Feng, Yuechen Wang, Wengang Zhou, Jiajun Deng, and Houqiang Li. 2022. Doctr: Document image transformer for geometric unwarping and illumination correction. *Preprint*, arXiv:2110.12942.

Jaekyu Ha, R.M. Haralick, and I.T. Phillips. 1995. Recursive x-y cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955 vol.2.

Leipeng Hao, Liangcai Gao, Xiaohan Yi, and Zhi Tang. 2016. A table detection method for pdf documents based on convolutional neural networks. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 287–292.

Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*.

Thomas Hegghammer. 2022. Ocr with tesseract, amazon textract, and google document ai: a benchmarking experiment. *Journal of Computational Social Science*, 5.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. *Preprint*, arXiv:2204.08387.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2021. Spatial dependency parsing for semi-structured document information extraction. *Preprint*, arXiv:2005.00642.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. *Preprint*, arXiv:1905.13538.

Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards understanding 2d documents. *Preprint*, arXiv:1809.08799.

Imed Keraghel, Stanislas Morbieu, and Mohamed Nadif. 2024. Recent advances in named entity recognition: A comprehensive survey and comparative study. *Preprint*, arXiv:2401.10825.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. *Preprint*, arXiv:2305.18486.

Chen-Yu Lee, Chun-Liang Li, Hao Zhang, Timothy Dozat, Vincent Perot, Guolong Su, Xiang Zhang, Kihyuk Sohn, Nikolai Glushnev, Renshen Wang, Joshua Ainslie, Shangbang Long, Siyang Qin, Yasuhisa Fujii, Nan Hua, and Tomas Pfister. 2023. Formnetv2: Multimodal graph contrastive learning for form document information extraction. *Preprint*, arXiv:2305.02549.

D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. 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*, SIGIR '06, page 665–666, New York, NY, USA. Association for Computing Machinery.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. 2023. Geolayoutlm: Geometric pre-training for visual information extraction. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chuwei Luo, Yufan Shen, Zhaoqing Zhu, Qi Zheng, Zhi Yu, and Cong Yao. 2024. Layoutllm: Layout instruction tuning with large language models for document understanding. *Preprint*, arXiv:2404.05225.

Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. 2020. Representation learning for information extraction from form-like documents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6495–6504, Online. Association for Computational Linguistics.

S. Marinai, M. Gori, and G. Soda. 2005. Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):23–35.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. *Preprint*, arXiv:2007.00398.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Jeju Island, Korea. Association for Computational Linguistics.

L. O'Gorman. 1993. The document spectrum for page layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1162–1173.

OpenAI. 2023. Gpt-4 is openai's most advanced system.

Rasmus Berg Palm, Ole Winther, and Florian Laws. 2017. Cloudscan - a configuration-free invoice analysis system using recurrent neural networks. *Preprint*, arXiv:1708.07403.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing.

Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, Shikun Feng, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2022. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. *Preprint*, arXiv:2210.06155.

Vincent Perot, Kai Kang, Florian Luisier, Guolong Su, Xiaoyu Sun, Ramya Sree Boppana, Zilong Wang, Zifeng Wang, Jiaqi Mu, Hao Zhang, Chen-Yu Lee, and Nan Hua. 2024. Lmdx: Language model-based document information extraction and localization. *Preprint*, arXiv:2309.10952.

Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. 2021. Going full-tilt boogie on document understanding with text-image-layout transformer. *Preprint*, arXiv:2102.09550.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

A. Simon, J.-C. Pret, and A.P. Johnson. 1997. A fast algorithm for bottom-up document layout analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):273–277.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2023a. Docllm: A layout-aware generative language model for multimodal document understanding. *Preprint*, arXiv:2401.00908.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023b. Gpt-ner: Named entity recognition via large language models. *Preprint*, arXiv:2304.10428.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, Jiazheng Xu, Bin Xu, Juanzi Li, Yuxiao Dong, Ming Ding, and Jie Tang. 2024. Cogvlm: Visual expert for pretrained language models. *Preprint*, arXiv:2311.03079.

Zifeng Wang, Zizhao Zhang, Jacob Devlin, Chen-Yu Lee, Guolong Su, Hao Zhang, Jennifer Dy, Vincent Perot, and Tomas Pfister. 2023c. Queryform: A simple zero-shot form entity query framework. *Preprint*, arXiv:2211.07730.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2022. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *Preprint*, arXiv:2012.14740.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20. ACM.

Zhibo Yang, Rujiao Long, Pengfei Wang, Sibo Song, Humen Zhong, Wenqing Cheng, Xiang Bai, and Cong Yao. 2023. Modeling entities as semantic points for visual information extraction in the wild. *Preprint*, arXiv:2303.13095.

Chong Zhang, Yi Tu, Yixi Zhao, Chenshu Yuan, Huan Chen, Yue Zhang, Mingxu Chai, Ya Guo, Huijia Zhu, Qi Zhang, and Tao Gui. 2024. Modeling layout reading order as ordering relations for visually-rich document understanding. *Preprint*, arXiv:2409.19672.

A BLOCKIE prompt templates

Table 6: Train Dataset Labeling Prompt Template

PROMPT INSTRUCTIONS
TAKE THE FOLLOWING TEXT - `<text>` THIS GETS PARSED INTO `<annotation>` BREAK THE PROVIDED TEXT INTO SEMANTIC BLOCKS LIKE $block_1, block_2...$ WITH **RELATED TEXT** IN SAME BLOCK. HERE ARE SOME RULES: 1/ THE OUTPUT SHOULD BE A DICTIONARY WITH KEYS - $block\_1, block\_2$ ETC. 2/ EACH BLOCK SHOULD BE A DICTIONARY ITSELF, WITH THE KEYS - REASON, TEXT AND PARSED: • IN REASON, THINK STEP-BY-STEP WHY THE TEXT UNDER CONSIDERATION IS A SINGLE BLOCK • THE TEXT KEY SHOULD CONTAIN THE TEXT PRESENT IN THE BLOCK • THE PARSED SECTION SHOULD CONTAIN THE PART OF THE PARSED OUTPUT THE TEXT MAPS TO 3/ RELATED TEXT REFERS TO TEXT BELONGING TO THE SAME <LINKED OR HIERARCHICAL ENTITY FROM SCHEMA, OR OTHERS> 4/ DO NOT LEAVE OUT ANY TEXT 5/ DO NOT WRITE A SINGLE EXTRA WORD

Table 7: Block Creator Prompt Template

PARSER INSTRUCTIONS

YOU ARE A SEASONED TEXT PARSER. GIVEN AN OCR TEXT, YOU ARE ABLE TO PARSE IT INTO BLOCKS OF RELATED TEXT ALONG WITH STEP-BY-STEP REASONS.

<Linked and Hierarchical entity identification rules>:
<Few Shot Examples> -
HERE ARE SOME RULES:
1/ THE OUTPUT SHOULD BE A DICTIONARY WITH KEYS - BLOCK_1, BLOCK_2 ETC.
2/ EACH BLOCK SHOULD BE A DICTIONARY ITSELF, WITH THE KEYS - REASON, AND TEXT.
  A. IN REASON, THINK STEP-BY-STEP WHY THE TEXT UNDER CONSIDERATION IS A SINGLE BLOCK. SHOW STEP BY STEP REASONING USING RULES AND EXAMPLES LAID OUT.
  B. THE TEXT KEY SHOULD CONTAIN THE TEXT PRESENT IN THE BLOCK.
3/ RELATED TEXT REFERS TO TEXT BELONGING TO THE SAME <LINKED OR HIERARCHICAL ENTITY FROM SCHEMA, OR OTHERS>
4/ DO NOT LEAVE OUT ANY TEXT.
5/ DO NOT WRITE A SINGLE EXTRA WORD.

<Verification Process>
COMPLETE THE ANSWER FOR THE FOLLOWING TEXT. DO NOT WRITE ANYTHING EXTRA.
<OCR words> <bounding boxes>
ANSWER:

Table 8: Block Parser Prompt Template

SYSTEM INSTRUCTIONS

YOU ARE AN EXPERT SYSTEM FOR PARSING RECEIPT TEXT BLOCKS INTO STRUCTURED DATA. YOUR ROLE IS TO ANALYZE RECEIPT TEXT AND CONVERT IT INTO A STRUCTURED DICTIONARY FORMAT.

<SCHEMA AND FIELD DESCRIPTIONS>
<Formatting rules>

SIMILAR EXAMPLES FOR REFERENCE:
<few shot examples>
NOTE: THESE EXAMPLES ARE FOR REFERENCE BUT MAY CONTAIN SOME INCONSISTENCIES. FOLLOW THE RULES ABOVE STRICTLY.

CURRENT TASK:
THIS IS A BLOCK CREATED PREVIOUSLY WHERE THE BLOCK-CREATOR HAD THIS REASON "{query_reason}"

YOUR TASK IS TO CREATE A COMPLETE, VALID JSON DICTIONARY FOLLOWING THE PROVIDED SCHEMA THAT REPRESENTS ALL THE INFORMATION IN THIS RECEIPT DOCUMENT.

<OUTPUT SPECIFICATION>
<Verification Process>

PARSE THIS RECEIPT BLOCK INTO THE SCHEMA FORMAT:
<query_block>

Table 9: Block Combiner Prompt Template

System Instructions

You are an expert system for parsing receipt documents into structured data. Your task is to analyze a complete receipt document and create a comprehensive dictionary using partial information from individual blocks.
 	
CONTEXT:
You will be provided with:
1. All the words in the document
2. Bounding boxes
3. Individual blocks of text and their partial parses
4. The required dictionary schema
 	
`<SCHEMA AND FIELD DESCRIPTIONS>`
`<Linked and Hierarchical entity identification rules>`
 	
ALL WORDS IN THE DOCUMENT:
`{text}`
 	
ALL BOUNDING BOXES IN THE DOCUMENT:
`{bboxes}`
 	
PARSED BLOCKS:
Below are the individual blocks and their partial parses along with reason. Use these to help construct the complete dictionary:
`{blocks_and_parses}`
 	
INSTRUCTIONS:
1. Use the complete document text to understand the full context
2. Utilize the partial parses from blocks to help construct the final dictionary - remember - the partial parses may not have full context
3. Ensure all information is correctly categorized according to the schema
4. Maintain consistency with numerical formats from the original text
 	
`<Verification Process>`
 	
Your final dictionary should contain two keys:
1. reason - justify step by step why you chose particular values. Use the reason from partial parses, check if it mentions exact match.
2. invoice - share the invoice dictionary
 	
Return only the final JSON dictionary without any additional explanation with proper format.

B Datasets and Benchmarks

B.1 Datasets

CORD Dataset CORD (Park et al., 2019) contains 1000 Indonesian receipts, divided into train, validation and test samples of size 800,100 and 100. Along with the images, CORD also contains crowdsourced labels, and OCR output with bounding boxes. 30 hierarchical entities are annotated manually under top-level entities menu, subtotal and total. The associated task is to assign the words in the OCR output to these entities. Performance is assessed using micro-F1 on entity prediction.

SROIE Dataset SROIE (Huang et al., 2019) dataset consists of scanned receipts from a variety of domains, such as retail, food, and services, split into 626 train and 347 test receipts. The dataset contains images, OCR output and annotations with labeled entities for Company Name, Date, Total Amount, and Address. We evaluate our approach on the information extraction task proposed in the paper. Performance is assessed using micro-F1 on entity prediction.

FUNSD Dataset The FUNSD Datatet (Jaume et al., 2019) contains 199 fully annotated images of forms sampled from the form type document of the RVL-CDIP dataset (Harley et al.). The dataset is split into 149 images in the training set and 50 in the testing set. The annotations consist of text with four keys - question, answer, header, and others, which is simplistic and do not represent meaningful entities. However, the annotations also contain linkages, forming meaningful question-answer pairs and groupings of these pairs under headers. We focus on the entity-linking task to evaluate the ability of our approach to extract meaningful relations.

B.2 LLMs and Benchmark approaches

We tested out BLOCKIE across 5 different LLMs from two different families. The LLMs chosen are widely used and vary in sizes from massive proprietary models to open-source models with 7B parameters.

Claude 3.5 Sonnet Claude 3.5 Sonnet is the first model released by Anthropic from the Claude 3.5 family (Anthropic, 2024a). In the benchmark evaluations released by Anthropic, it showed at-par or superior performance compared to Claude 3 Opus, the previous best-performing Anthropic model, while being 2x faster. It established new state-of-the-art on reasoning and question-answering tasks at the time of its release.

For few-shot Sonnet results, we conducted experiments using the CORD validation dataset and found best results when 5 examples were used that were the closest (with respect to text embedding similarity) to the target sample.

Qwen 2.5 Qwen 2.5 is a family of open-source LLMs released by Alibaba Cloud (Qwen et al., 2025). The family contains both base language models, instruction-tuned models as well as specialized models for coding, math, etc. The family consists of models in sizes varying from 0.5B parameters to 32B parameters. We used the 72B, 32B, 14B and 7B versions for our experimentation.

For finetuning, we used LORA (Hu et al., 2021) with rank 64 for 6 epochs with learning rate 0.00002. These numbers were based on results obtained on the validation dataset of CORD.

LayoutLMV3 LayoutLMV3 (Huang et al., 2022) is a state-of-the-art information extraction benchmark. It incorporates layout information using cross-attention between bounding boxes and text, and through masked image modeling. It shows competitive performance on all three benchmark datasets. Note that while (Luo et al., 2023) outperforms LayoutLMV3, the authors have not officially released their pre-processing code or fine-tuned weights for CORD. We use layoutlmv3 in our experiments to demonstrate the limitations of SER-based approaches.

When we finetuned LayoutLMV3 for our experiments on heterogeneity and value-absent inference, we used the parameters listed in the official paper for CORD.

We reviewed the licenses for all these datasets and models, and ensured that we stick to the intended usage of these for research purposes.