An agentic system with reinforcement-learned subsystem improvements for parsing form-like documents

Ayesha Amjad^{1[0009-0002-4806-7207]}, Saurav Sthapit^{2[0000-0002-7421-0479]}, and Tahir Qasim Syed^{1[0000-0003-0638-9689]}

Abstract. Extracting alphanumeric data from form-like documents such as invoices, purchase orders, bills, and financial documents is often performed via vision (OCR) and learning algorithms or monolithic pipelines with limited potential for systemic improvements. We propose an agentic AI system that leverages Large Language Model (LLM) agents and a reinforcement learning (RL) driver agent to automate consistent, selfimproving extraction under LLM inference uncertainty. Our work highlights the limitations of monolithic LLM-based extraction and introduces a modular, multi-agent framework with task-specific prompts and an RL policy of rewards and penalties to guide a meta-prompting agent to learn from past errors and improve prompt-based actor agents. This self-corrective adaptive system handles diverse documents, file formats, layouts, and LLMs, aiming to automate accurate information extraction without the need for human intervention. Results as reported on two benchmark datasets of SOIRE, and CORD, are promising for the agentic AI framework.

Keywords: agentic \cdot multi-agent \cdot foundation models \cdot reinforcement learning \cdot form-like documents

1 Introduction

A significant volume of form-like documents [14] is generated every day by several industry verticals ¹, which includes invoices, purchase orders, receipts, bank statements, bills, and more. These documents require information extraction [23] to serve downstream tasks and applications in structured formats, such as efficient archiving, fast indexing, and document analytics. However, both traditional and contemporary approaches to extracting structured information from these documents are limited in their capability to efficiently deal with varying layouts, formats, and complexity [14].

¹ Industry verticals include retail, finance, healthcare, insurance, manufacturing, and businesses in general

Contemporary usage of deep learning algorithms such as convolutional neural networks (CNN) [8], [1] and graph neural networks (GNN) [29], [7], was especially influential for detecting tabular structure by leveraging its graphical properties. Methods proposing recurrent neural networks (RNN) [28], and transformer architecture for key information extraction [26], [13], [31], [14] suggested spatial relationship, and semantic entity recognition as primary contributing factors. However, these models require large training data and boast complex architectures. Additionally, CNNs and GNNs focus primarily on the layout analysis to perform extraction, ignoring semantics and contextual awareness. Overall these models assume a list of predefined entities, hence, struggle with unseen documents and entities.

Recent studies have discovered remarkable propensity of human-like reasoning [20], planning [12], judgment, and rationality [34] in LLMs, leading to the possibility of building applications with autonomous decision-making and execution called agents [33]. To solve the complex problem of automated data extraction effectively, a system of multiple LLM agents working in a collaborative environment while learning from past experiences is presented in this study. The evaluation within the framework is done through self-feedback systems [22] as well as metrics to compute matches and similarity between raw data and its extracted counterpart. Self-learning and optimization of each LLM agent is driven through Gymnasium [30], a reinforcement learning framework, where action space is in a natural language setting.

1.1 Understanding Form-Like Documents

In form-like documents, critical information resides as key-value pairs and line items. A Key-value pair is a pair of linked data items, where a key is the unique identifier to look up the value [14], and is largely drawn from a small vocabulary of field-specific variants [23]. Line items are a list of repeated instances of items typically detailing transactional information [19] that may or may not have graphical borders.

Variations in these data artifacts are found across different document types. For example, a key phrase "Invoice#" can also be written as "Invoice Number" or "Invoice No." and its value can consist of either all digits or an alpha-numeric sequence. Moreover, a key phrase and corresponding value can be positioned relative to each other, e.g., a value can be found in the same line as the key phrase with an in-between separator such as a colon (:), or directly underneath. Certain key fields such as "Date" are found in all form-like documents, but most of the other fields are unique to the document type.

Traditionally, detecting line items on a text file has been a complex problem due to variations in its layout and the inability of models to "classify" a structure as tabular based on its content alone [19]. Computer vision algorithms using Hough transform [1] have been successful in detecting graphical borders, but challenges persist for tables without borders.

2 Related Work

Convolutional neural networks such as Fast R-CNN with Region Proposal Networks [8], and encoder-decoder architectures like Chargrid [19] were effective for table and line items detection but struggled with ambiguous or boundary-less table layouts. [1] proposed YOLOv5 for table detection and predefined keyword searches for key-value extraction. Graph-based approaches enhanced performance by encoding spatial and semantic relationships; for instance, GNNenriched node embeddings differentiate between tables, headers, and plain text in [7]. RNN-based systems such as CloudScan [24] aided key-value extraction for invoices but were limited by their sequential processing and difficulty in generalizing to unseen layouts. Advances in representation learning [23] and contrastive learning, particularly in FormNetV2 [21], improved adaptability by encoding neighboring context and pretraining on form structure, although they still assumed predefined entities.

Transformer-based and LLM-driven methods highlights the efficacy for layoutand language-agnostic information extraction. Layout-aware transformers such as XYLayoutLM [11], spatial transformers like BROS [13], and semantic-rich models like TILT [26] and LiLT [32] significantly outperformed earlier deep learning models. LiLT, in particular, enabled multilingual document parsing. GenIE [18] autoregressively generates structured triplets (subject, relation, object), while KPVFormer [14] used a Q&A-based encoder-decoder architecture to extract key-value pairs. LLM-based frameworks offer a paradigm shift—methods such as trigger-based zero-shot extraction [3], instruction-tuned LLaMA via LoRA [17], and multimodal fine-tuned GenIK [2] demonstrate state-of-the-art flexibility, usability, and robustness, especially in low-resource or OCR-imperfect settings.

Existing models are trained on monolingual documents, except LiLT [32], finetuned for specific document categories like invoices, or assume a predefined list of entities. This limits their effectiveness with unseen documents and dynamic layouts. Their complex architectures and sensitivity to hyperparameters also hinder widespread adoption. Furthermore, standardized evaluation metrics to report the extraction accuracy and system confidence is essential for all probabilistic models in automated environments.

3 Methodology

The implemented framework (Figure 1) introduces a novel agentic approach to automated document data extraction by combining reinforcement learning with LLMs. As detailed in Section 4, this framework is evaluated against a baseline using both proprietary data and public benchmarks, including CORD [25] and ICDAR-SOIRE [15]. GPT-40-mini [16] serves as the default LLM, and LLaMA 3.3-70B [6] is an alternative for processing sensitive or confidential documents. The token limit adheres to each model's maximum capacity to handle lengthy inputs, and all responses are constrained to a strict JSON format for structured output.



Fig. 1. Architecture of the agentic unstructured data extraction framework comprising seven agents (blue), two Gymnasium environments (dark pink), and five evaluation metrics (cyan). Agents include a document classifier, schema builder, data extractor, two Gymnasium agents, a meta-prompting agent, and an evaluator. Evaluation metrics guide iterative optimization of the environments.

3.1 Document Text Reader

A document reader class reads text off of form-like documents in various formats (PDF, TXT, DOC, PNG, JPG, JPEG, TIFF, BMP), while preserving the layout of the original document. For scanned PDFs and images, it uses state-of-the-art paddle OCR [4], ensuring accurate conversion into machine-readable text along with a built-in confidence score. Text-based formats (PDF, TXT, DOC) bypass OCR for downstream processing.

To preserve the original layout of documents post-OCR, we first compute the center position of each text block using $x_{center} = \frac{\min(x_1, x_2, x_3, x_4) + \max(x_1, x_2, x_3, x_4)}{2}$. Layout preservation is then guided by a vertical threshold $\Delta y_{threshold} = h_{page} \cdot 0.015$, which helps identify line breaks and group related text blocks. To maintain appropriate spacing between elements, we calculate discrete block spacing as $\max\left(1, \left\lfloor \frac{x_{current} - x_{last}}{w_{page} \cdot 0.01} \right\rfloor\right)$. These heuristics enable the reconstruction of structured layouts while accommodating document variability.

3.2 Evaluation Metrics

Extraction accuracy is evaluated using two non-parametric scores - exact match and cosine similarity. JSON output is corroborated against its ground-truth for exact matches, calculated as a simple ratio of intersection to union. It serves as a proxy for accuracy, helping validate extracted values and identifying hallucination in an LLM-generated output.

$$Exact Match = \frac{number_of_exact_matches}{total fields}$$

Cosine similarity score is computed between the generated JSON (A) and ground-truth JSON (B). It is robust against partial matches, and tolerant to schema variability.

Cosine Similarity =
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The agentic process generates JSON schema dynamically for each document, leading to deviation in field and header names from its ground-truth and resulting in zero exact-match scores. To address this, a semantic match metric—using LLM-based evaluation but same scoring formula as exact match—is introduced.

Semantic match includes two variants: a basic version with binary field-level comparison and simple score aggregation, and an enhanced version with row alignment and cell accuracy:

 $\label{eq:overall Score} \mbox{Overall Score} = f(\mbox{confidence_score}, \mbox{table_analysis}, \mbox{field_analysis})$

Table Score =
$$\frac{\text{row}_\text{alignment} + \text{header}_\text{match} + \text{cell}_\text{accuracy}}{3}$$

Classical machine learning metrics of F1, precision, and recall are used particularly for a comparative analysis on benchmark datasets against current stateof-the-art methods.

3.3 Baseline Data Extraction Framework

The process begins by reading text off of a form-like document, and appending it with a static "system" prompt. A real-time API call is made to the LLM, similar to a sequence-to-sequence task of Q&A [14], with upto 3 retries triggered by a predefined evaluation thresholds. The "system" prompt provides chain-ofthought instructions to establish context-aware extraction, and "user" prompt contains the source document text. The user prompt contains metadata specifying the page number, and document type (e.g. invoice, purchase order, utility bills). This contextual information helps the LLM understand the structure and semantics of the document.

4 Agentic Data Extraction Framework

The agentic framework is designed to adapt to any document type, format, and LLM. The system employs a Gymnasium-compatible environment that models the extraction process as a Markov Decision Process (MDP) [27]. It comprises of five fundamental components: actor agents, Gymnasium environments, observed evaluation space, meta-prompting [9] action space (4.4), and RL agents (4.5). These components are essential building blocks of the self-directed and self-corrected multi-agent extraction pipeline shown in figure 1.

4.1 Document Classification Agent

Document classification initiates the analysis of the document's first page using an LLM to categorize it into predefined classes: Invoice, Purchase Order, Utility Bill, Receipt, Financial Document (e.g., bank statements, balance sheets, investment reports), Salary Slip, or Unknown (fallback).

The classification confidence is approximated through a probabilistic approach, where the LLM's logarithmic probabilities are transformed into linear probability by taking an exponential of logprobs as shown below:

$$P_{\text{linear}} = e^{\text{logprob}} \cdot 100\%$$

Documents either failing to meet minimum P_linear thresholds or a known category are flagged as 'unknown' for a manual review, thereby maintaining system reliability and accuracy.

4.2 Automated Schema Building

The schema building stage implements an iterative interaction between five components of agentic framework. The custom Gymnasium environment supports an episodic learning framework for schema actor prompt with the following components:

State Representation: Two-dimensional continuous space S_s , representing perplexity score $p \in [0, \infty)$, and schema complexity $c \in [0, 1]$. Perplexity scores are computed by exponentiating the negative of the average of the logprobs. It captures the uncertainty of the generated schema, with no upper bound, its lower values indicate higher schema quality.

$$\mathbf{p} = \exp\left(-\frac{1}{n}\sum_{i=1}^{n} log probs\right)$$

Schema complexity is a computed score between 0 and 1. It measures the structural complexity, where scores closer to 0 indicate simpler, more maintainable schema.

$$\mathbf{c} = \alpha N + \beta D + \gamma B + \delta R$$

where, N is the nesting depth, D is data type diversity. B is the branching factor, and R is the reference complexity. The weights are constrained such that:

$$\alpha + \beta + \gamma + \delta = 1$$

where: $\alpha = 0.4, \beta = 0.2, \gamma = 0.2, \delta = 0.2.$

Reward Function: Reward is computed linearly. The reward function $R_i(s_t, a_t)$ for state s_t and action a_t at time horizon t uses perplexity score p_t and complexity score c_t . Total reward is a simple aggregation of individual step rewards r_i .

$$R_i(s_t, a_t) = (p_{\text{best}} - p_t) + (c_{\text{best}} - c_t)$$

Action Space: The action space A consists of a discrete set of five prompt engineering strategies, $A = \{0, 1, 2, 3, 4\}$, each optimizing the prompts as per instructions (see 4.4).

Termination Criteria: The schema building process continues until either maximum steps are reached, or no improvements are observed in consecutive iterations.

 $terminated = \begin{cases} 1 & \text{if steps} \ge \max_steps \\ 1 & \text{if non_improvement_count} \ge 2 \\ 0 & \text{otherwise} \end{cases}$

The dual optimization approaches (perplexity and complexity) provide complementary perspectives on schema quality, maintaining consistent improvement trajectories, and ensuring robustness while maintaining adaptability.

4.3 Iterative Data Extraction Environment

The multi-agent framework adopts an iterative approach, enabling interactions between the data extractor agent, extraction environment, meta-prompting action space, and an RL policy for optimizing actor prompts. It dynamically refines extraction strategies based on self-feedback .

Multi-page extraction is supported via a worker pool (Section 4.7), where pages are handled in parallel and results are concatenated into the final output JSON.

As part of the experiment, the framework implements three distinct environment variants with episodic learning:

- 1. DataExtractionEnvBase: Basic implementation with simple multiplicative reward mechanisms and fixed thresholds-based termination conditions.
- 2. DataExtractionEnvIterative: Enhanced version using performance plateaus P(t) to implement adaptive termination T(s,t), and improvement-based with bonus reward system $R(s_t, a_t, s'_t)$.
- 3. DataExtractionEnvStepCount: Step-limited version with explicit exploration or exploitation trade-offs with time-penalization.

Following architecture discusses the second environment *DataExtractionEnvIterative* as the best performing framework for agentic data extraction.

State Representation: The state space S is defined as a continuous 3-dimensional vector. S is tracked over time horizon T, with best performing metrics and corresponding best prompt and output. For any time step $t \in T$:

$$t^* = \underset{t \in T}{\arg \max} (s_{exact}^t + s_{semantic}^t + s_{similarity}^t)$$
$$\mathcal{B} = \left\{ \underset{t \in T}{\max} s_{exact}^t, \underset{t \in T}{\max} s_{semantic}^t, \underset{t \in T}{\max} s_{similarity}^t, \omega^{t^*}, \pi^{t^*} \right\}$$

where, s^t are the scores at time t, and (ω, π) represent the output-prompt pair.

Reward Function: A step function for a combined score $\sigma_t = s_{exact} + s_{semantic} + s_{similarity}$, and an improvement bonus coefficient $\beta = 0.1$

$$R(s_t, a_t, s'_t) = \begin{cases} \beta + \sigma_t - \sigma_{t-1} & \text{if } \sigma_t > \mathcal{B} \\ \sigma_t - \sigma_{t-1} & \text{otherwise} \end{cases}$$

Termination Criteria: The iterative environment implements adaptive termination [T(s,t)] based on performance plateaus P(t) achievement and non-improvement tracking:

$$P(t) = \begin{cases} 1 & \text{if } \sigma_t \leq \sigma_{t-1} \text{ for } k \text{ consecutive steps} \\ 0 & \text{otherwise} \end{cases}$$

where, k = 2 (configurable non-improvement threshold)

$$T(s,t) = \begin{cases} 1 & \text{if } P(t) = 1\\ 1 & \text{if steps} \ge \max_\text{steps}\\ 1 & \text{if threshold_conditions_met}(s)\\ 0 & \text{otherwise} \end{cases}$$

4.4 Meta-Prompting Agent

The meta-prompting agent represents a systematic approach to dynamic prompt engineering, whereby, using driver prompts to refine and optimize actor prompts. These meta-prompting strategies makes up the discrete action space $A \in \{0, 1, 2, 3, 4\}$:

$$A = A(s, p, t, o, g) = \begin{cases} Bt(p) & \text{if } a = 0\\ C(p) & \text{if } a = 1\\ F(p, t_k) & \text{if } a = 2\\ N(p) & \text{if } a = 3\\ R(p, F(p, o, g)) & \text{if } a = 4 \end{cases}$$

- 1. Best Practice Strategy (Bt(p)): Optimize prompts by letting LLM decide and establish best prompt engineering practices.
- 2. Clarity Enhancement Strategy (C(p)): Applies process decomposition, ambiguity elimination, tone optimization, structural enhancement, and goal clarification.
- 3. Few-Shot Learning Strategy $(F(p, t_k))$: Implements example-based learning through carefully curated demonstrations for each task type t_k . A maximum of 3 examples are included to prevent context overflow.
- 4. Feedback-Refine Optimization Strategy (R(p, F(p, o, g))): Takes inspiration from LLM-as-a-judge [10] policy, where an LLM evaluates and critics for complex tasks. It is a 2-stage process, where first a comprehensive feedback is provided by observing actor prompt p, generated output o and corresponding ground truth g. Followed by refinements in actor prompt to minimize differences between generated and target output.
- 5. Preservation Strategy (No Change) N(p): Used as a control experiment that maintains original actor prompt.

4.5 Reinforcement Learning Agent

The Gymnasium agent implements a language model-guided policy $\pi(a|s)$ that observes the current state vector, current total rewards, and a boolean value for task completion, analyzes it through an LLM and maps to a meta-prompting action. This creates a closed-loop optimization system where prompt improvements are guided by both immediate feedback and long-term performance metrics.

$$\pi(a|s): S \to P(A)$$

where, P(A) is a probability distribution over action space A.

The system maintains state tracking to maximize the expected cumulative reward R for all iterative environments with minimum terminal step T:

$$argmax(\mathbf{E}[\sum_{t=0}^{T} \gamma^{t} R])$$

$$T = \min(T_{\max}, \inf\{t : n_t \ge N\})$$

where, T_{max} is the maximum allowed steps, n_t is the number of consecutive non-improvements, and N is the non-improvement threshold.

Exploration rate decay $\epsilon(t)$ of Gymnasium RL policy states:

$$\epsilon(t) = \epsilon_0 \cdot e^{-\delta}$$

where, $\epsilon_0 = 1.0$ (initial exploration rate), and $\delta = 0.2$ (decay rate)

4.6 Learned Prompt Optimization

Learned Prompt Optimization (LPO) extends the capability of a simple RL agent by leveraging contextual bandit learner to iteratively explore and exploit effective meta-prompting strategies. It comprises of three core components:

- 1. Vector Embedding: Converts textual inputs into dense vector representations using OpenAI ada-002 embedding model.
- 2. Selection Scorer: Evaluates prompt strategies based on context and historical performance.
- 3. Policy Learner: Updates strategy selection weights using Vowpal Wabbit [5] contextual bandit implementation.

For each iteration t, the probability of selecting action $a_i \in A$ is:

$$P(a_i|C_t) = \frac{\exp(\theta_i^T \phi(C_t))}{\sum_{i=1}^n \exp(\theta_i^T \phi(C_t))}$$

where $\phi(C_t)$ is the context embedding and θ_i are learned parameters. After observing cumulative reward R, parameters are updated as follows, where η is the learning rate.

$$\theta_i \leftarrow \theta_i + \eta R \nabla_{\theta_i} \log P(a_i | C_t)$$

4.7 System Optimization Components

The agentic data extraction framework encounters operational challenges with multi-page documents. A single page requires time t with multiple LLM calls for optimized output, hence, processing n pages scales linearly to nt, leading to significant computational overhead. Additionally, concatenating text from all n pages into a single LLM call, while seems feasible, dampens output accuracy.

To overcome these challenges and build a robust document processing system with minimal redundant operations, parallel processing and caching mechanisms are implemented. Furthermore, our sequential framework, where the output from each step influences subsequent steps, requires comprehensive logging and error handling for debugging and monitoring to maintain system reliability.

5 Results

Results are reported for 32665 form-like documents, including 20295 proprietary documents and 11000 benchmark files, out of which approximately 93% requires an OCR processing. The largest volume of document type is invoice with 17492 files, followed by 12987 receipts. Most documents comprises of 2-3 pages (Average Number of Pages = $\frac{\text{Total Number of Pages}}{\text{Number of Files}} = 2.7$).

5.1 Baseline Framework Results

Table 1 reports the baseline results of the LLM-generated output on 32665 files, via one-shot prompting, on two metrics - exact match, and cosine similarity. All reported scores are best of 3 pooled responses averaged over # of files for each document type. Two factors have been observed to influence the evaluation performance, 1) Number of pages, 2) Scan quality.

Document Type	Formats	# of Files	Exact Match	Similarity
Invoices	Images	10,000	0.43	0.61
Invoices (Australian)	Digital PDF	2202	0.58	0.74
Invoices (Australian)	Scanned PDF	5290	0.33	0.38
Purchase Orders	Digital PDF	19	0.78	0.76
Purchase Orders	Scanned PDF	138	0.43	0.55
Receipts	Images	987	0.78	0.81
Financial documents	Mixed/HTML	2,832	0.30	0.59
Utility Bills	Images	100	0.31	0.75
Salary Slips	Images	97	0.84	0.86
CORD	Images	10,000	0.71	0.90
ICDAR-SROIE	Images	1,000	0.67	0.75

Table 1. Results from the one-shot single-execution prompt

Table 2 extends the analysis by reporting on classical machine learning metrics of F1, *precision*, and *recall* for benchmark datasets.

11

Document Type	F1 Score	Precision	Recall
CORD	0.886	0.902	0.871
ICDAR-SROIE	0.808	0.780	0.838

 Table 2. Results from baseline framework on two benchmark datasets

Financial documents, typically spanning 7 or more pages and available in high-quality formats, yield low extraction scores (30% exact matches, 59% similarity). In contrast, single-page images of receipts at 100 dpi (standard threshold = 300 dpi)—achieve higher scores (78% exact, 81% similarity). Similarly, single-page utility bills with even lower quality (89 dpi) still outperform financial documents, but falls behind receipts, in extraction accuracy (See figure 2).



Fig. 2. a) Extracted JSON of a receipt (96 dpi) scored 90% on exact matches, 100% on semantic matches, and 96% on cosine similarity. b) Extracted JSON of a utility bill (66 dpi) scored lower: 30% exact matches, 65% semantic matches, 71% similarity.

Two patterns are observed in the extracted JSON of multi-page documents; partial extraction across all pages (often missing line items from middle pages), and complete omission of entire pages (more common). Multiple reruns revealed inconsistent outputs, suggesting nondeterministic behavior from the LLM. The nondeterministic behavior worsens with poor quality scans, which sometimes returned a blank JSON or incorrect (See figure 3) and hallucinated response.



Fig. 3. Incorrect data extracted for a 2-page scanned document. "3093" is extracted as "goog" and "4155" is extracted as "4is5"

Limitations of static prompts without explicit output schema are evident in cases with atypical key phrases or headers. Cases discussed in figure 4 highlights the importance of including output schema in the extraction prompts. Extraction accuracy is influenced not just by page count and scan quality, but also by document formatting. Multi-line headers and dark-shaded key phrases often reduce accuracy, mainly due to issues in text conversion rather than LLM behavior. Processing documents as images generally yields more reliable results than pre-converting them to text.



Fig. 4. Top: 'Invoice Total' is misidentified as 'total' due to semantic closeness. Bottom: explicitly specifying 'CODE' as a column header enables correct extraction, even though there is no 'CODE' column header in the document table.

The baseline monolithic prompt-based system exhibit nondeterministic behavior, producing inconsistent outputs with multi-page documents, tables with missing headers, poor quality scans, ambiguous field names, and atypical document formatting. Context window limitations further exacerbate the issue, restricting processing to shorter documents. There is also a higher likelihood of hallucination resulting from information density of multi-page documents.

5.2 Agentic Framework Results

Table 3 reports the results of agentic data extraction framework on three evaluation metrics. The exact match scores are computed after a manual verification of each output JSON against its groundtruth, therefore, depicting the most accurate picture of the output performance. Semantic match is used to balance out a score of zero on the exact match during runtime iterative processing.

mini		
Document Type	Formats	# of Files Exact Match Semantic Match Sim

Table 3. Results from the self-directed and self-corrected multi-agent system - *qpt-4o*-

Document Type	Formats	# of Files	Exact Match	Semantic Match	Similarity
Invoices (mixed)	Images	10,000	0.863	0.901	0.941
Invoices (Australian)	Searchable PDF	2202	0.985	0.944	0.908
Invoices (Australian)	Scanned PDF	5290	0.913	0.900	0.939
Purchase Orders	Searchable PDF	19	0.994	1.00	0.980
Purchase Orders	Scanned PDF	138	0.953	0.899	0.952
Receipts	Images	987	0.834	0.910	0.866
Financial documents	Images	2,832	0.962	0.998	0.927
Utility Bills	Images	100	0.425	0.663	0.817
Salary Slips	Images	97	1.00	0.998	0.959
CORD	Images	10,000	0.866	0.812	0.921
ICDAR-SROIE	Images	1,000	0.753	0.911	0.890

A clear improvement is observed in extraction completeness from every page of financial documents. For instance, if a bank statement page lists 15 transactions, all are accurately captured in the JSON, regardless of its page index or document length. This boosts the exact match scores from 30% to 96.2% and similarity scores from 59% to 92.7%, primarily due to multi-page handling where each page is processed independently.

Searchable PDFs or high resolution single-page images achieve near-perfect extraction completeness and accuracy. Purchase Orders show a semantic score of 1.00, exact match of 0.994, and cosine similarity of 0.98. Similarly, each salary slip achieves a 100% exact match and ranks only second to Purchase Orders on other metrics.

The agentic process remains robust against large file sizes and complex structure. When a document contain multiple tables, the algorithm extracts each as a distinct child collection in the JSON output without overly complicating the schema. As shown in figure 5, even with duplicate tables, the schema builder agent filters out redundancies to retain only unique instances for extraction.

(iii) /iii preving Content Annuali, Proping Microsoft (Content Annuali, Proping). (iii) Second and Second and Second and Second and Second Annualist Second and Second Annualist Second	Karon Art Studio 1134 Main Strut Louisville, KV 40211 (502) 125-1234	Smith, John 1123 River 8 Louisville, K	and Y 42911	SSN Geoss Pay Nat Pay Filling Status	XXX-3X-3123 \$1,288.46 \$999.79 M3	Pariod Beginning Pariod Ending Check Date Check Na	6/3/2000 6/17/2010 6/17/2010 1000	"metadata": ("SSW": "string", "Employee Name": "string", "Employee Maders", "string",
1. "Untract Netabers": Directly and extract only the netabers of key-robse pairs and table hashers from the desinent.	Tarrines	Hoers/Rate	Amount	YTD Ann	Taxa Delectors	Amount	YTD Am	"Gross Pay": "number".
 ""Structure the ISSN Scheme": - "Struct Registr": List all key phrases with their corresponding data taxes. 	Yeles		\$1.285.46	\$1.288.06	Fed Income Tax	\$104.23	\$164.23	"Period Beginning": "date",
- ""Dills Region": List all table hadens with their corresponding data types.	Outing	0.0	\$3.00	\$0.00	Social Security Tax	\$75.78	\$56.78	"Period Ending": "date",
1. "Multials deter": bears that the order of my same and table bashes matches the original document.	Banus		\$8.00	\$0.00	Modeare Tax State Income Tax	\$17.94	\$17.96 \$64.70	"Net Pay": "number",
 "Volume Extrine") Delate only unline online mess and try strates, available dualitation. 	Commission		50.00	\$3.02	Local Incente Tax	50.00	\$8.00	"Check Date": "date",
	Green Pay		\$1,288.46	\$1,288.45	Health Insurance	\$15.00	\$15.00	"Filing Status": "string",
 "Multiple Tables": D' the document consults multiple tables, represent each table in the our parent region ubtion the 35% orient. 					401K	\$50.00	\$58.00	"Chark No": "integer"
6. "Neverse logo": If a value lado a definal lag or basier, crusts an appropriate lag or basier for that value.					Not Pay	\$459.75	\$859.79	"YTD Amount": "number"
7. ""http://formet": Augord exclusively with the 200 scheme containing all key phrases and balls colum headers.								}.
## togle:								"Earnings": (
Mongle 1/1								"Salary": "number",
Document ("Overtime": "number",
lass: http://www.international.com								"Bonus": "number",
Ages 100								"Comissions": "number".
inspation Eginer	Karen Art Studio	Smith, John		SSN	XXX-XX-3123	Period Beginning	6/3/2000	"Time"s Combos"
Table: Indexs	1134 Main Street	1123 River 8	laad	Ceors Pay	\$1,788.46	Pariod Ending	6/1/2010	TAPS - Dumper
- Anglett Name	Louisville, KY 40211	Lourville, 8	4.0011	No Pay	3979.19	Check lower	1000	1.
- 5041.0404 - 604 Nota	0023123-1234			ring states	10	0.0000.000		"Deductions": {
	Kaminar	Hours Weitr	Amount	YTD Area	Times Deductions	Amoun	YID Ami	"Federal Income Tax": "number"
200 Scheral	-comp							"Social Security Tay": "number
- jan	Silary		\$1,288.45	\$1,288.46	Fed Income Tax	\$164.23	\$164.23	Social Secondry lax . House
The second se	Overline	0.0	\$3.00	\$0.00	Social Security Tat	s 556.71	\$36.78	medicare lax : number ,
Twent Dividg's	Broat		\$3.00	\$0.00	Modicare Tax	\$17.94	\$17.55	"State Income Tax": "number",
Tap's Temper's	Curaniziona		\$3.00	\$0.00	State Income Tax	\$64.78	564.79	"Local Income Tax": "number"
and and a strong	Tips		\$3.00	\$0.00	Lecol lecome Tax	50.00	\$0.00	Thesiah Terresonal Company
Windows ()	Gress Per		\$1,288.46	\$1,258.46	Health Insurance	\$15.00	\$15.00	meator thsurance : "number",
"Windows News's "sectory".					401K	\$56.0	\$\$3.00	"481K": "number"
"Start Bear": "Wea".					Nat Pay	\$\$55.7	\$ \$859.79	1
the same is made								

Fig. 5. This salary slip shows four tables, but only two are unique, as earnings and tax deductions each appear twice. Schema builder prompt (left) retains only unique tables.

Low-quality scans (≤ 100 dpi) remain challenging due to the framework's reliance on OCR, as evident by the poor performance of utility bills scoring lowest across all metrics. A potential solution is to bypass OCR and use multimodal LLMs for direct image-based data extraction.

The agentic process dynamically generates and optimizes the schema at runtime, enabling it to extract nearly all relevant fields—often beyond the scope of benchmark groundtruth. In figure 6, the extracted JSON of a receipt includes all expected fields except one (address) and also captures additional fields not present in the groundtruth. To accommodate schema variations between the groundtruth and extracted output, a flexible evaluation using fuzzy matching is employed for F1, precision, and recall (Table 4). Values are considered a match if their similarity is $\geq 80\%$, allowing minor formatting differences to be ignored.

tan woon yann	O x0001640818_gpt_extracted.jon × D II ···	8 100016405019an ×
Control Control Control State State State State State S	<pre>cl concerning.security * 0 = 0 =</pre>	# (second second secon
Thank You I Please Love Again 1 Deads Bold Fro BOL Returnation Basing In Ubicals And Potati.	27 1 28 1 29 1	

Fig. 6. ICDAR-SROIE data collection (left), agentic extracted JSON (center), groundtruth (right)

Table 4. Agentic results on benchmark shows a minimum 0.05 point boost on classical ML metrics $% \left({{\rm{Table}}\;{\rm{A}}} \right)$

Document Type	F1 Score	Precision	Recall
CORD	0.965	0.966	0.964
ICDAR-SROIE	0.939	0.958	0.921

Integrating document classification and schema building stages creates a fully automated agentic extraction pipeline. Table 5 reports on classification confidence (linear probability) and schema evaluation scores—averaged per document type. These metrics are as critical in evaluating the agentic workflow.

Document Type	Confidence	Accuracy	Best Complexity
Invoices	98.327%	0.84	0.547
Purchase Orders	98.196%	1.00	0.620
Receipts	87.482%	0.910	0.288
Financial documents	98.292%	0.928	0.434
Utility Bills	98.5%	0.903	0.727
Salary Slips	1.00%	1.00	0.488

 Table 5. Evaluation report on document classification and schema generation stages

Salary slips are classified with 100% confidence, while some financial documents and utility bills are misidentified as 'unknown'. This is likely due to classification relying only on the first page, which may be a cover page in financial documents, and the low-resolution images typical of utility bills. Invoices exhibit lower classification accuracy due to false positives, as many receipts are incorrectly predicted as invoices.Schema complexity is lowest for receipts due to its simple structure, and highest for utility bills. The schema building component improves the extraction but introduces variability, as each schema is tailored per document. Documents with similar layouts but different content may produce different metadata, which is acceptable for extraction but problematic for downstream ETL processes that require uniform structures.



Fig. 7. Baseline vs Agentic - comparison on exact matches

As illustrated in figure 7, comparing baseline vs agentic frameworks, where agentic shows improvements across all document types with financial documents and Australian invoices reporting massive increase of more than 0.5 point in

exact match scores. Same is the case for all other evaluation metrics, concluding that the agentic process has shown remarkable potential to the form-like data extraction problem in an automated environment.

6 Conclusion

Single prompt driven applications, leveraging LLMs, have shown promising results in simple data extraction tasks. These applications are effortless, fast, and produces quality output for single-page documents. However, it faces inherent limitations in handling complex, multi-page unstructured documents. Common challenges include limited context window, hallucination due to information density, and predefined prompts faltering on varying documents. This constrains the reliability and scalability of monolithic preset-prompt frameworks.

The challenges are mitigated by Agentic systems. Modular, multi-agent architectures distribute tasks and improve robustness. Systems deal inaccuracies by leveraging specialized agents for classification, splitting, schema generation and adaptive extraction. These improvements makes the process ideal for midsize documents of good resolution. Moreover, their ability to provide confidence metrics and iterative feedback ensures higher accuracy and scalability, making them an essential evolution for form-like document data extraction.

6.1 Limitations

In addition to OCR reliance, we have observed that meta-prompting is sensitive to the wordings of the initial prompt. Processing speed is another challenge as agentic takes a minimum of 1 minute to extract data, while the baseline takes less than 10 seconds on average. The system caters to only a handful of document categories, and there is no automated way of handling documents classified as 'unknown'. Future studies may explore multimodality of LLMs for image data extraction, in addition to adding more learnable parameters for RL policy.

6.2 Ethical Statements

The usage of cloud-based LLMs to process documents containing critical information has been under ethical scrutiny since the release of OpenAI gpt-3.5. The risks include data privacy, potential security breaches, unauthorized access, and compliance with regulatory frameworks like GDPR and HIPAA.

All data used in this study is handled in accordance with the best practices of encryption and access control. The research utilizes OpenAI's enterprise API, which is designed to ensure a secure and compliant processing. Moreover, evaluation scores are reported on benchmark datasets to compare the performance of the system with existing state-of-the-art methodologies, to avoid making confidential proprietary data public.

References

- 1. Arslan. Arslan, H.: End to End Н., Invoice Processing Appli-Based on Key Fields Extraction. IEEE Access (2022).cation https://doi.org/10.1109/access.2022.3192828
- Cao, P., Wang, Y., Zhang, Q., Meng, Z.: GenKIE: Robust Generative Multimodal Document Key Information Extraction. Conference on Empirical Methods in Natural Language Processing (2023). https://doi.org/10.48550/arxiv.2310.16131
- Cao, R., Rongyu, C., Cao, R., Luo, P., Luo, P., Luo, P.: Extracting Zero-shot Structured Information from Form-like Documents: Pretraining with Keys and Triggers. AAAI Conference on Artificial Intelligence (2021). https://doi.org/10.1609/aaai.v35i14.17494
- 4. contributors, P.: Paddleocr: Awesome multilingual ocr toolkits based on paddlepaddle. https://github.com/PaddlePaddle/PaddleOCR, accessed: 2024-12-08
- 5. Contributors, V.: Vowpalwabbit reinforcement learning (2024), accessed: 2025-04-20
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
- Gemelli, A., Vivoli, E., Marinai, S.: Graph Neural Networks and Representation Embedding for Table Extraction in PDF Documents. International Conference on Pattern Recognition (2022). https://doi.org/10.1109/icpr56361.2022.9956590
- Gilani, A., Gilani, A., Qasim, S.R., Qasim, S.R., Malik, I., Malik, I., Shafait, F., Shafait, F.: Table Detection Using Deep Learning. IEEE International Conference on Document Analysis and Recognition (2017). https://doi.org/10.1109/icdar.2017.131
- 9. Goodman, N.: Meta-prompt: A simple self-improving ai design (2023), https://noahgoodman.substack.com/p/meta-prompt-a-simple-self-improving
- 10. Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, Y., Guo, J.: A survey on llm-as-a-judge. ArXiv abs/2411.15594 (2024), https://api.semanticscholar.org/CorpusID:274234014
- Gu, Z., Meng, C., Wang, K., Lan, J., Wang, W., Gu, M., Zhang, L.: XY-LayoutLM: Towards Layout-Aware Multimodal Networks For Visually-Rich Document Understanding. Computer Vision and Pattern Recognition (2022). https://doi.org/10.1109/cvpr52688.2022.00454
- Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X.: Large language model based multi-agents: A survey of progress and challenges (2024), https://arxiv.org/abs/2402.01680
- Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., Park, S.: BROS: A Pretrained Language Model Focusing on Text and Layout for Better Key Information Extraction from Documents. AAAI Conference on Artificial Intelligence (2021). https://doi.org/10.1609/aaai.v36i10.21322
- Hu, K., Wu, Z., Zhong, Z., Lin, W., Sun, L., Huo, Q.: A Question-Answering Approach to Key Value Pair Extraction from Form-like Document Images. AAAI Conference on Artificial Intelligence (2023). https://doi.org/10.48550/arxiv.2304.07957
- Huang, Z., Huang, Z., Chen, K., Chen, K., Chen, K., Jiang, H., He, J., Bai, X., Bai, X., Karatzas, D., Karatzas, D., Karatzas, D., Karatzas, D., Lu, S., Lu, S., Jawahar, C.V., Jawahar, C.V.: ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. IEEE International Conference on Document Analysis and Recognition (2019). https://doi.org/10.1109/icdar.2019.00244

- 18 A. Amjad et al.
- Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
- Jiao, Y., Zhong, M., Li, S., Zhao, R., Ouyang, S., Ji, H., Han, J.: Instruct and Extract: Instruction Tuning for On-Demand Information Extraction. Conference on Empirical Methods in Natural Language Processing (2023). https://doi.org/10.48550/arxiv.2310.16040
- Josifoski, M., Josifoski, M., Cao, N.D., Cao, N.D., Peyrard, M., Peyrard, M., Petroni, F., Petroni, F., West, R., West, R.: GenIE: Generative Information Extraction. North American Chapter of the Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.naacl-main.342
- Katti, A.R., Katti, A.R., Katti, A.R., Reisswig, C., Reisswig, C., Reisswig, C., Reisswig, C., Guder, C., Guder, C., Brarda, S., Brarda, S., Bickel, S., Bickel, S., Höhne, J., Höhne, J., Faddoul, J.B., Faddoul, J.B., Faddoul, J.B., Faddoul, J.B.: Chargrid: Towards Understanding 2D Documents. Conference on Empirical Methods in Natural Language Processing (2018). https://doi.org/10.18653/v1/d18-1476
- Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. Advances in neural information processing systems 35, 22199–22213 (2022)
- 21. Lee, C.Y., Li, C.L., Zhang, H., Dozat, T., Pérot, V., Su, G., Zhang, X., Sohn, K., Glushnev, N., Wang, R., Ainslie, J., Long, S., Qin, S., Fujii, Y., Hua, N., Pfister, T.: FormNetV2: Multimodal Graph Contrastive Learning for Form Document Information Extraction. Annual Meeting of the Association for Computational Linguistics (2023). https://doi.org/10.48550/arxiv.2305.02549
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B.P., Hermann, K., Welleck, S., Yazdanbakhsh, A., Clark, P.: Self-refine: Iterative refinement with self-feedback (2023), https://arxiv.org/abs/2303.17651
- Majumder, B.P., Majumder, B.P., Majumder, B.P., Potti, N., Potti, N., Tata, S., Tata, S., Wendt, J.B., Wendt, J.B., Wendt, J.B., Zhao, Q., Zhao, Q., Najork, M., Najork, M., Najork, M.: Representation Learning for Information Extraction from Form-like Documents. Annual Meeting of the Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.580
- Palm, R.B., Winther, O., Laws, F.: CloudScan A configuration-free invoice analysis system using recurrent neural networks (Aug 2017), http://arxiv.org/abs/1708.07403, arXiv:1708.07403 [cs]
- Park, S.H., Park, S., Park, S., Shin, S.H., Shin, S.Y., Shin, S., Shin, S.H., Shin, S.K., Lee, B., Lee, B., Lee, B., Lee, J., Lee, J., Lee, J., Surh, J., Surh, J., Surh, J., Surh, J., Seo, M., Seo, M., Lee, H., Lee, H.S.: CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. null (2019). https://doi.org/null
- Powalski, R., Powalski, R., Powalski, R., Borchmann, , Borchmann, L., Jurkiewicz, D., Jurkiewicz, D., Dwojak, T., Dwojak, T., Pietruszka, M., Pietruszka, M., Pałka, G., Pałka, G.: Going Full-TILT Boogie on Document Understanding with Text-Image-Layout Transformer. IEEE International Conference on Document Analysis and Recognition (2021). https://doi.org/10.1007/978-3-030-86331-9 47
- 27. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., USA, 1st edn. (1994)
- Rasmus Palm, Rasmus Berg Palm, Ole Winther, Ole Winther, Florian Laws, Florian Laws: CloudScan - A Configuration-Free Invoice Analysis System Using Recurrent Neural Networks. IEEE International Conference on Document Analysis and Recognition (2017). https://doi.org/10.1109/icdar.2017.74

19

- Riba, P., Riba, P., Dutta, A., Dutta, A., Goldmann, L., Goldmann, L., Fornés, A., Fornés, A., Ramos, O., Ramos, O., Lladós, J., Lladós, J.: Table Detection in Invoice Documents by Graph Neural Networks. IEEE International Conference on Document Analysis and Recognition (2019). https://doi.org/10.1109/icdar.2019.00028
- Towers, M., Kwiatkowski, A., Terry, J., Balis, J.U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., KG, A., et al.: Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint arXiv:2407.17032 (2024)
- Wang, J., Jin, L., Ding, K.: LiLT: A Simple yet Effective Language-Independent Layout Transformer for Structured Document Understanding. Annual Meeting of the Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.acl-long.534
- 32. Wang, J., Jin, L., Ding, K.: LiLT: A Simple yet Effective Language-Independent Layout Transformer for Structured Document Understanding. Annual Meeting of the Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.acl-long.534
- 33. Woodridge, M., Jennings, N.: Intelligent agents: Theory and practice the knowledge engineering review. The Knowledge Engineering Review (1995)
- Xu, L., Hu, Z., Zhou, D., Ren, H., Dong, Z., Keutzer, K., Ng, S.K., Feng, J.: Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration (2024), https://arxiv.org/abs/2311.08562