SAIL: Sample-Centric In-Context Learning for Document Information Extraction

Jinyu Zhang[1]*, Zhiyuan You[2]*, Jize Wang[1], Xinyi Le[1]†

[1]Shanghai Jiao Tong University
[2]The Chinese University of Hong Kong
{zhang_jinyu, jizewang2000, lexinyi}@sjtu.edu.cn, zhiyuanyou@foxmail.com

Abstract

Document Information Extraction (DIE) aims to extract structured information from Visually Rich Documents (VRDs). Previous full-training approaches have demonstrated strong performance but may struggle with generalization to unseen data. In contrast, training-free methods leverage powerful pre-trained models like Large Language Models (LLMs) to address various downstream tasks with only a few examples. Nonetheless, training-free methods for DIE encounter two primary challenges: (1) understanding the complex relationship between layout and textual elements in VRDs, and (2) providing accurate guidance to pre-trained models. To address these challenges, we propose SAmple-centric In-context Learning (SAIL). SAIL introduces a fine-grained entity-level textual similarity to facilitate in-depth text analysis by LLMs and incorporates layout similarity to enhance the analysis of layouts in VRDs. Moreover, SAIL formulates a unified In-Context Learning (ICL) prompt template for various sample-centric examples, enabling tailored prompts that deliver precise guidance to pre-trained models for each sample. Extensive experiments on FUNSD, CORD, and SROIE benchmarks with various base models (*e.g.*, LLMs) indicate that our SAIL outperforms training-free baselines, even closer to the full-training methods, showing the superiority and generalization of our method.

Code — https://github.com/sky-goldfish/SAIL

1 Introduction

Document Information Extraction (DIE) focuses on extracting structured information from Visually Rich Documents (VRDs) such as receipts, forms, and invoices (Park et al. 2019; Huang et al. 2019; Jaume, Ekenel, and Thiran 2019). Previous works, including LayoutLMv3 (Huang et al. 2022), primarily concentrate on full-training methodologies that demand extensive task-specific labeled data. While these models have achieved notable success on the trained dataset, they often struggle to generalize effectively to unseen data, especially when the test data distribution significantly diverges from that of the training data. To address this challenge, training-free DIE methods (He et al. 2023) leverage powerful pre-trained models like Large Language Models (LLMs) that can generalize to unseen data given only a few examples, and thus begin to attract more research interests.

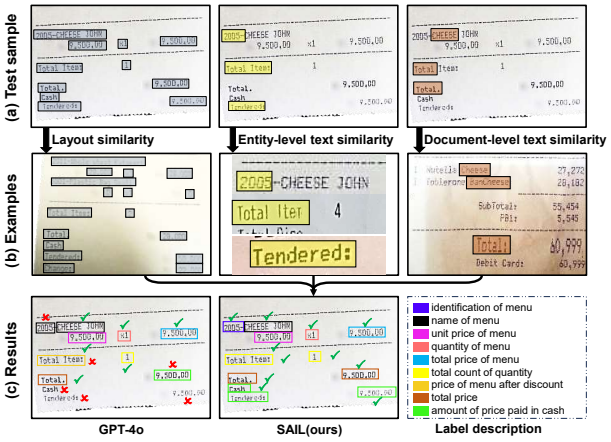

Figure 1: For the (a) test sample from the CORD dataset (Park et al. 2019), our SAIL selects (b) layout similarity examples (grey marked), entity-level similarity examples (yellow marked), and document-level similarity examples (orange marked) to construct ICL prompts. (c) Benefiting from these examples, SAIL precisely extracts all information, while even the powerful GPT-4o (OpenAI 2023b) misidentifies three entities and incorrectly labels three entities.

One of the primary challenges in the training-free DIE task is understanding the complex relationship between the document layout and its textual entities using only a few examples. VRDs possess discrete textual elements alongside flexible, inherently structured layouts, complicating the establishment of relationships between textual entities and the extraction of implicit layout information. Even the advanced multi-modal LLMs like GPT-4o (OpenAI 2023b) demonstrate limited effectiveness in performing DIE task. As illustrated in Figure 1(c), GPT-4o misidentifies three entity texts and labels three entity texts incorrectly, highlighting the challenges inherent in the training-free DIE task.

Another significant challenge is providing a clear and effective guidance to pre-trained models (*e.g.*, LLMs). Although these models possess extensive knowledge and capa-

*These authors contributed equally.
†Corresponding author

bilities, they necessitate appropriate instructions for optimal performance on specific downstream tasks. Recent research has incorporated In-Context Learning (ICL) within LLMs to enhance the DIE performance (He et al. 2023). This approach involves selecting a few textually similar examples and carefully crafting the in-context prompts with diverse demonstrations for the entire dataset. While this method shows promising results in GPT-3.5 (Brown et al. 2020), the fixed in-context examples fail to effectively guide different LLMs, leading to a significant performance decline when transitioning across different LLMs, as detailed in Table 1.

To address these challenges, we propose a SAmple-centric In-context Learning (SAIL) method. Our method follows two core principles: (a) To enhance LLMs' understanding of the complex interplay between layout and text within VRDs, the provided prompts must analyze the question from different angles in depth. (b) To ensure precise guidance, it is essential to develop a customized prompt for each test sample. Regarding the first principle, previous methods (He et al. 2023) only adopted rough document-level textual similarity for example selection, which inadequately supports LLMs in understanding textual information in lengthy documents. Consequently, we propose a refined entity-level text similarity for in-depth text analysis. Additionally, we incorporate layout similarity to identify examples that enjoy similar layouts, facilitating LLMs in comprehending complex layout information in VRDs. The three distinct examples are illustrated in figure 1(b). For the second principle, we select distinct examples for each test sample and integrate them into a unified prompt template with clear instructions to devise a tailored sample-centric in-context prompt.

Equipped with these designs, our proposed SAIL demonstrates versatility across various LLMs on multiple benchmarks. SAIL not only stably surpasses all training-free baselines, but even achieves comparable performance to many fully-trained models when implemented with GPT-4. Overall, our main contributions can be summarized as follows:

- We introduce **layout similarity** and **entity-level text similarity**, each highlighting unique facets of VRDs, resulting in a thorough and in-depth analysis of VRDs.
- To form **sample-centric** in-context prompts, we propose a unified ICL prompt template applicable to various examples. With clear instructions, LLMs enhance their attention to specific information in the examples.
- We conduct extensive experiments on multiple benchmarks including FUNSD, CORD, and SROIE with various base LLMs. Our SAIL achieves superior performance than training-free baselines, even closer to the performance of full-training methods.

2 Related Works

Document Information Extraction (DIE). Traditional DIE methods primarily rely on extensive datasets for model pre-training and subsequent fine-tuning on downstream tasks. These methods can be classified into four main categories. The first category consists of grid-based methods (Katti et al. 2018; Zhao et al. 2019; Denk and Reisswig 2019; Kerroumi, Sayem, and Shabou 2021), which encode each document page as a two-dimensional character grid of characters to preserve the document's layout. The second category, graph-based methods, utilizes Graph Convolutional Networks (GCN) (Qian et al. 2019; Liu et al. 2019) or Graph Neural Networks (GNN) (Tang et al. 2021) for DIE. The third category encompasses transformer-based (Vaswani et al. 2017) methods. Traditional methods design small models in specialized fields. Some methods integrate text semantics and layout modality for model pre-training (Li et al. 2021; Hong et al. 2022; Wang, Jin, and Ding 2022; Wang et al. 2023b), while other methods jointly leverage text, layout, and image modality to enhance document understanding (Da et al. 2023; Xu et al. 2020, 2021; Huang et al. 2022). A recent trend has seen numerous studies employing LLMs' advanced language capabilities. (Wang et al. 2023a; Perot et al. 2023; Lu et al. 2024; Li et al. 2024; Luo et al. 2024; Fujitake 2024). In contrast to the categories above that necessitate OCR for text and box recognition, the final category aims to bypass the OCR process and establish end-to-end models (Wang et al. 2021; Kim et al. 2022; Liu et al. 2024b; Mao et al. 2024; Hu et al. 2024; Abramovich et al. 2024). Despite the notable performance of many methods, they demand retraining for specific downstream tasks.

In-Context Learning (ICL). Brown et al. (2020) discovered that pre-trained LLMs can address unseen tasks using only a few examples without weight updates through ICL. From then on, ICL has been widely adopted in question answering (Yang et al. 2022; Liu et al. 2023; Wang et al. 2024), multi-modal named entity recognition (Cai et al. 2023), and dialogue improvement (Meade et al. 2023; Hu et al. 2022).

ICL-based DIE. ICL presents a viable approach for performing the DIE task with minimal examples. ICL-D3IE (He et al. 2023), the first work to construct ICL prompts for DIE, utilizes diverse demonstrations through examples selected via text semantic search. Nonetheless, ICL-D3IE exhibits limited generalization capabilities to novel LLMs, primarily due to its reliance on fixed examples and handcrafted prompts. Our method clearly distinguishes itself from this work. First, we dynamically select unique examples for each test sample, in contrast to ICL-D3IE's fixed examples. Second, we employ a unified template to construct prompts that can be generalized to various LLMs, while ICL-D3IE adopts specifically designed prompts that are less adaptable to new models. Third, we demonstrate that relying solely on document-level text similarity is inadequate for identifying optimal examples, and thus introduce layout similarity and entity-level text similarity for enhanced performance. With these designs, our method achieves better results than ICL-D3IE across various base LLMs.

3 Methods

3.1 Problem Formulation

Training-free DIE leverages pre-trained models (*e.g.*, LLMs) to extract specified categories of text information (*e.g.*, company, address, and date (Huang et al. 2019)) from VRDs. Specifically, given a document image I, the goal is to label all entities within I. First, entity texts $T = \{t_1, t_2, ..., t_{n_e}\}$ and their corresponding boxes $B =$

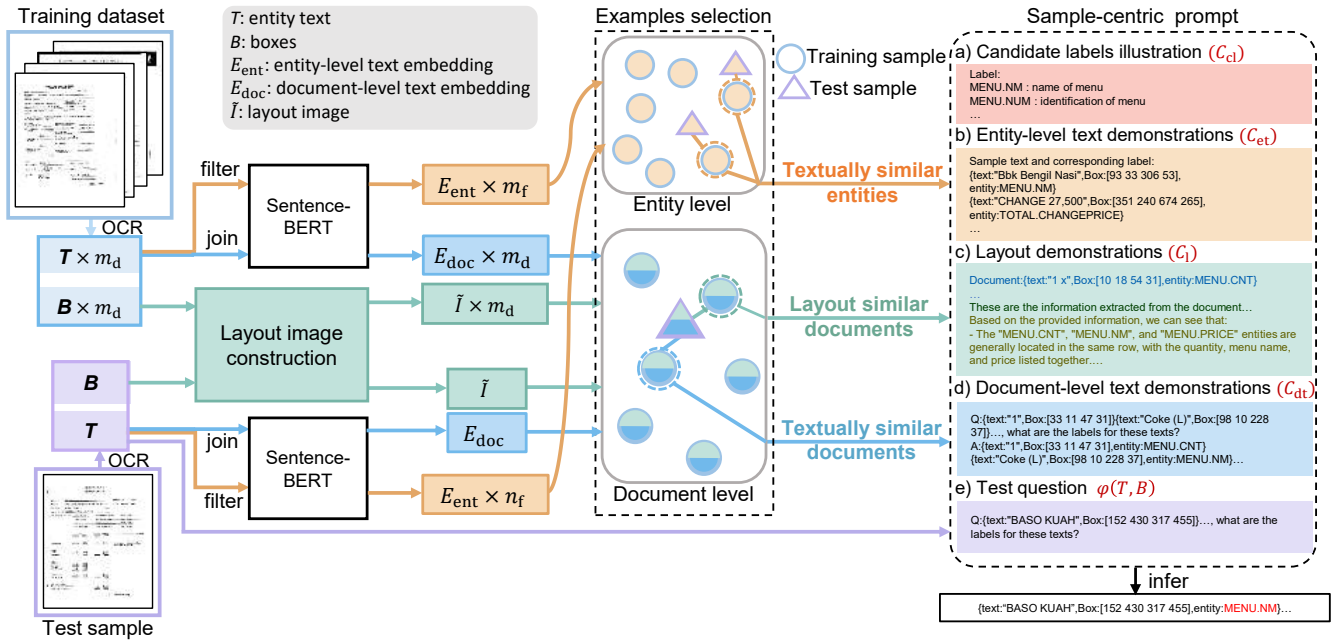

Figure 2: **Illustration of SAIL framework**, including extracting texts T and boxes B from document images, encoding them separately, selecting textually similar entities, layout similar documents, and textually similar documents for each test sample, constructing sample-centric prompts using diverse examples, and generating predicted labels.

$\{b_1, b_2, ..., b_{n_e}\}$ are recognized from I by an OCR system, where n_e is the total number of entities in the document image. To effectively utilize LLMs, in-context prompts C are designed to convey the extraction intention. For ICL-based DIE, C is constructed by selecting several examples demonstrating how to solve DIE tasks. With these in-context prompts as illustrations, LLMs are tasked with generating labels Y_{pred} for all detected entities. The process is achieved by maximizing the conditional probability $P(Y|T, B)$ while incorporating the prompts C as an additional condition:

$$P(Y|T,B) = \frac{1}{n_e}\sum_{k=1}^{n_e} P_{LM}(l_k|C, \varphi(T,B)), \quad (1)$$

where P_{LM} is the conditional probability predicted by the LLMs, and φ denotes the operation of converting the entity texts and boxes into a textual format suitable for LLMs' input. In training-free DIE, the construction of effective in-context prompt C is crucial, which is the primary focus of this work. Finally, the predicted labels Y_{pred} are evaluated using F1 scores against the ground truth labels Y_{gt}.

3.2 Overview Framework

To maximize $P(Y|T, B)$ with the in-context prompt C, we propose SAIL, a sample-centric in-context prompt construction method for DIE. SAIL focuses on designing C for individual samples by automatically selecting tailored layout examples, document-level text similarity examples, and entity-level text similarity examples based on the test sample, subsequently leveraging these examples to generate C.

The overall architecture, illustrated in Figure 2, comprises five steps. Firstly, the test document image and m training document images are processed through OCR to extract entity texts T and boxes B. Secondly, T are transformed into entity-level text embeddings E_{ent} and document-level text embeddings E_{doc}. B are used to construct layout image $\tilde{I}$. Thirdly, E_{ent}, $\tilde{I}$ and E_{doc} are used to select textually similar entities, layout similar documents, and textually similar documents for the test sample. Then, these selections are substituted into the prompt template to form a tailored in-context prompt C. Finally, LLM performs inference with C and question $\varphi(T, B)$ to generate predicted labels Y_{pred}.

3.3 Document-Level Text Similarity Examples

To improve the capability of ICL, we employ text semantic search to select the nearest training document examples for a given test sample (Liu et al. 2022). The entity texts T extracted from a document image are concatenated into a single sentence and encoded with Sentence-BERT (Reimers and Gurevych 2019), resulting in a text semantic embedding E_{doc} for the document. We determine the nearest training examples by computing the document-level text similarity T_{sim_doc} between the test embedding E^{test}_{doc} and m training embeddings E^{train}_{doc} using the cosine similarity score:

$$T_{sim\_doc} = \frac{E^{test}_{doc} \cdot E^{train}_{doc}}{||E^{test}_{doc}||\ ||E^{train}_{doc}||}. \quad (2)$$

3.4 Entity-Level Text Similarity Examples

The document-level text similarity T_{sim_doc} between a lengthy text document and the found text-similar documents

Figure 3: **Illustration of layout similarity evaluation**, including drawing boxes onto a blank image, cropping and resizing to form layout image, and comparing layout images.

is notably low. To facilitate LLMs in generating text with more relevant examples for learning, we propose entity-level text similarity examples, as shown in Figure 2.

Entity texts $T = \{t_1, t_2, \ldots, t_{n_e}\}$ recognized by OCR are filtered to exclude texts consisting solely of numbers, which provide minimal semantic content. Subsequently, the filtered m_f training entity texts and n_f test entity texts are encoded using Sentence-BERT to derive the semantic embedding E_{ent}. The entity-level text similarity $T_{\text{sim\_ent}}$ is computed from the semantic embedding E_{ent} by employing the cosine similarity score, defined as follows:

$$T_{\text{sim\_ent}} = \frac{E_{\text{ent}}^{\text{test}} \cdot E_{\text{ent}}^{\text{train}}}{||E_{\text{ent}}^{\text{test}}||\ ||E_{\text{ent}}^{\text{train}}||}. \tag{3}$$

We select n_s textually similar entities for each test entity by nearest neighbor search and obtain $n_f \times n_s$ examples.

3.5 Layout Similarity Examples

To identify documents with similar layouts, we introduce a layout similarity assessment methodology, illustrated in Figure 3. Firstly, all b_i from boxes $B = \{b_1, b_2, ..., b_{n_e}\}$ are rendered as black rectangles on a blank image. Subsequently, we define the information area as the minimal region that contains all entity texts and crop the layout image to maintain a 10-pixel margin between the information area and the image borders. Next, we standardize the layout image dimensions through resizing. Finally, we select n_s layout similar documents by calculating the layout similarity L_{sim} between the training layout image $\tilde{I}^{\text{train}}$ and the test layout image $\tilde{I}^{\text{test}}$ using Mean Square Error (MSE) loss:

$$L_{\text{sim}} = \frac{1}{\text{MSE}} = \frac{n_l}{(U-V)^{\text{T}}(U-V)}, \tag{4}$$

where U, V are the pixel matrix of $\tilde{I}^{\text{train}}$ and $\tilde{I}^{\text{test}}$, and n_l is the total number of pixels in the layout image.

Moreover, to enhance the understanding of layouts by LLMs, we substitute the boxes from the cropped image B' for all documents in the prompt instead of using B.

3.6 Sample-Centric ICL Prompt Template

To construct C for an individual test sample, we propose an adaptive sample-specific ICL prompt template. The template is comprised of 5 parts: candidate labels illustration C_{cl}, entity-level text demonstrations C_{et}, layout demonstrations C_{l}, document-level text demonstrations C_{dt} and test question $\varphi(T, B)$, as shown on the right of Figure 2.

Candidate labels illustration C_{cl} enumerates all potential labels for the DIE task. For abbreviated labels, a corresponding natural language description is appended.

Entity-level text demonstrations C_{et} present textually similar entities. The prompt p_{e} "*Sample text and corresponding label:*" in conjunction with the labels of the selected n_s textually similar entity examples Y_{et}, formulates the entity-level text similarity demonstrations:

$$C_{\text{et}} = \text{CONCAT}[p_{\text{e}}, Y_{\text{et}}]. \tag{5}$$

Layout demonstrations C_{l} aim to facilitate LLMs in analyzing the layout of the test document. After obtaining n_s layout similar documents, we introduce a layout analysis step. This step enables LLMs to comprehend the overall document structure and the relationship between layout and label selection. The layout analysis prompt p_{a} is defined as: "*These are the information extracted from the document through OCR, and the Box is the position of the text in the document. Please analyze where each label is generally located in the document.*", which can apply to any dataset. The labels of layout-similar documents Y_{l} are input into LLMs together with p_{a}, allowing LLMs to analyze the layout information in layout-similar documents by themselves. The resulting output from the LLM is denoted as A_{l}. A layout similarity demonstration C_{l} is formulated as follows:

$$C_{\text{l}} = \text{CONCAT}[Y_{\text{l}}, p_{\text{a}}, A_{\text{l}}]. \tag{6}$$

Document-level text demonstrations C_{dt} showcase textually similar documents in question-answer format, guiding LLMs to produce answers in a specific format. The textually similar documents X_{dt}, the ground truth answer Y_{dt} and the DIE instruction p_{q} such as "*What are the labels for these texts?*" form the Document-level text demonstration prompt:

$$C_{\text{dt}} = \text{CONCAT}[X_{\text{dt}}, p_{\text{q}}, Y_{\text{dt}}]. \tag{7}$$

Finally, the test question $\varphi(T, B)$ for the test sample is:

$$\varphi(T, B) = \text{CONCAT}[T, B', p_{\text{q}}]. \tag{8}$$

3.7 Inference

After selecting a diverse set of examples, ICL prompts facilitate LLMs in generating entity labels Y_{pred}. This process is mathematically represented as follows:

$$P(Y|T, B) = \frac{1}{n_e} \sum_{k=1}^{n_e} P_{\text{LM}}(l_k | C_{\text{cl}}, C_{\text{et}}, C_{\text{l}}, C_{\text{dt}}, \varphi(T, B)). \tag{9}$$

Subsequently, entity labels Y_{pred} are extracted from the generated output. We assess the accuracy of Y_{pred} against the ground truth labels Y_{gt} utilizing the F1 score.

4 Experiments

4.1 Datasets, Metrics, and Details

FUNSD (Jaume, Ekenel, and Thiran 2019) is a dataset for understanding the content of tables in scanned documents.

It contains 149 tables and 7,411 entities in the training set, and 50 tables and 2,332 entities in the test set. In the DIE task, the candidate labels of the FUNSD dataset include "Header", "Question", "Answer", and "Other".

SROIE (Huang et al. 2019) is another scanned receipt understanding dataset, containing 626 receipts in the training set and 347 in the test set. The DIE task needs to extract "company", "date", "address", and "total" information.

CORD (Park et al. 2019) is a receipt understanding dataset that contains 800 training data, 100 test data, and 100 validation data. This dataset features 30 detailed and hierarchical labels, much more than the above two datasets.

Metrics. Following previous works (He et al. 2023), we adopt entity-level **F1 score**, **precision** and **recall** as metrics.

Details. We evaluate our method using three LLMs: the open-source ChatGLM3 (THUDM 2023) and the closed-source GPT-3.5 (OpenAI 2023a) and GPT-4 (OpenAI 2023b). Specifically, we use the `chatglm3-6b-32k` version for ChatGLM3, `gpt-3.5-turbo` API version for GPT-3.5, and `gpt-4o` API version for GPT-4. For GPT-3.5 and GPT-4o, we set the temperature parameter to 0 to enhance the reproducibility. In the case of GPT-4o, we only provide text prompts as input, while also testing its multi-modal capabilities by providing document images and clear task instructions. In our experiments, for each test document, we select four textually similar documents and four layout-similar documents as examples due to the limitation of prompt token number. Furthermore, for each filtered test entity, we choose four textually similar entity examples.

4.2 Results on DIE Benchmarks

Baselines. We compare our SAIL against baseline models including BERT (Devlin et al. 2019), LiLT (Wang, Jin, and Ding 2022), BROS (Hong et al. 2022), XYLayoutLM (Gu et al. 2022), LayoutLM (Gu et al. 2022), LayoutLMv2 (Xu et al. 2021), and LayoutLMv3 (Huang et al. 2022) in both full-training and few-shot settings. We borrow their metrics from (He et al. 2023). Training-free methods including standard ICL and ICL-D3IE (He et al. 2023) are also compared. ICL-D3IE only reports the performance of standard ICL and ICL-D3IE with GPT-3.5, so we evaluate their performance with GPT-4 and ChatGLM3 using their official repositories.

Quantitative results are presented in Table 1. First, overall, our method stably outperforms ICL-D3IE across different LLMs on all datasets. Second, when switching the LLM from GPT-3.5 to ChatGLM3, the performance drop of ICL-D3IE is significantly larger than our SAIL (*e.g.*, -73.8% *vs.*-12.73% in SROIE), demonstrating that our method has better robustness and adaptability to various LLMs. Third, the performance of ICL-D3IE degrades slightly when transitioning from GPT-3.5 to the more advanced GPT-4 on FUNSD and SROIE datasets, further indicating its incompatibility with new LLMs. However, in all datasets, our method achieves better performance on more advanced GPT-4 than on GPT-3.5, which is intuitive and reasonable. These results demonstrate the advantages of our method.

Qualitative results are illustrated in Figure 4. ICL-D3IE incorrectly predicts the entities on the three left green boxes as "answer", while our SAIL accurately identifies them as "question". This indicates that fixed examples in ICL-D3IE are insufficient to guide LLMs in effectively learning the relationship between discrete texts, highlighting the importance of selecting diverse examples for each test sample.

Setting	Methods		FUNSD	CORD	SROIE
Full-Training	$\text{BERT}_{\text{BASE}}$		60.26	89.68	90.99
	$\text{LiLT}_{\text{BASE}}$		88.41	96.07	94.68
	$\text{BROS}_{\text{BASE}}$		83.05	95.73	95.48
	$\text{XYLayoutLM}_{\text{BASE}}$		83.35	94.45	95.74
	$\text{LayoutLM}_{\text{BASE}}$		79.27	91.06	94.38
	$\text{LayoutLMv2}_{\text{BASE}}$		82.76	94.95	96.25
	$\text{LayoutLMv3}_{\text{BASE}}$		90.29	96.56	96.89
Few-Shot	$\text{BERT}_{\text{BASE}}$		38.76	38.88	38.76
	$\text{LiLT}_{\text{BASE}}$		54.88	69.12	84.03
	$\text{BROS}_{\text{BASE}}$		59.46	72.78	76.78
	$\text{XYLayoutLM}_{\text{BASE}}$		65.44	69.16	75.66
	$\text{LayoutLM}_{\text{BASE}}$		32.49	40.19	76.79
	$\text{LayoutLMv2}_{\text{BASE}}$		71.42	65.71	81.81
	$\text{LayoutLMv3}_{\text{BASE}}$		70.67	70.13	79.13
Training-Free	ChatGLM3	Standard ICL	40.93	67.30	81.37
		ICL-D3IE	35.90	36.44	18.83
		SAIL (ours)	**58.24**	**83.04**	**85.03**
	GPT-3.5	Standard ICL	72.76	68.34	82.11
		ICL-D3IE	**83.66**	87.13	92.63
		SAIL (ours)	83.48	**95.80**	**97.76**
	GPT-4	Standard ICL	75.15	90.22	96.00
		ICL-D3IE	78.94	87.47	89.23
		SAIL (ours)	**84.67**	**96.41**	**98.18**

Table 1: **Quantitative results** with F1 metric. Our SAIL stably surpasses baselines across various base LLMs.

4.3 Comparison with Multi-modal LLMs

Baselines. Recent years have witnessed the rapid development of multi-modal LLMs (MLLMs) represented by GPT-4o (OpenAI 2023b). To further validate the effectiveness of our method, we also compare our SAIL with MLLMs including open-source LLaVA-1.5 (Liu et al. 2024a) and proprietary GPT-4o. We provide these MLLMs with explicit and detailed instructions to inform the task definition.

Quantitative results are provided in Table 2. The open-source LLaVA exhibits limited DIE capabilities, resulting in a low F1 score (e.g., 0.7% in FUNSD). The proprietary GPT-4o significantly outperforms LLaVA (50.72% vs 0.7% in FUNSD), yet still falls short when compared to specialized DIE methods. Therefore, despite their rapid evolution, MLLMs still underperform in the DIE task, highlighting the importance and contribution of our proposed work.

4.4 Ablation Studies

Effect of Adaptive Example. We assess the influence of adaptive examples by employing both fixed and adaptive examples to construct in-context prompts within the same prompt template. The base LLM is selected as GPT-3.5, and the results are illustrated in Table 3. The utilization of adaptive examples results in superior F1 scores, confirming the

Methods	SROIE			CORD			FUNSD		
	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
GPT-4o	47.49	46.77	48.24	71.53	82.96	62.87	50.72	73.01	38.85
LLaVA-v1.5-7B	2.32	5.49	1.47	8.85	67.39	4.74	0.70	61.54	0.35
SAIL (ours)	**98.18**	**97.72**	**98.64**	**96.41**	**96.41**	**96.41**	**84.67**	**84.67**	**84.67**

Table 2: **Performance comparison** with multi-modal LLMs. Multi-modal LLMs even powerful GPT-4o still struggle with DIE tasks and our method significantly surpasses GPT-4o and LLaVA-v1.5-7B.

Setting	FUNSD	CORD	SROIE
Fixed example	74.23	82.35	91.08
Adaptive example	**83.48**	**95.80**	**97.76**
Δ	9.25	13.45	6.68

Table 3: **Ablation study of adaptive examples** with F1 metric. Adaptive examples is superior than fixed examples.

Text \ Layout	Ascending	Descending
Ascending	95.19	94.73
Descending	94.73	**95.80**

Table 4: **Performance comparison of the example order** in the prompt with F1 metric in the CORD dataset.

#	Similar Text-Doc.	Similar Layout	Similar Text-Ent.	FUNSD	CORD	SROIE
0				69.87	83.04	95.08
1	✓			69.60	92.13	96.38
2	✓	✓		73.13	92.97	97.24
3	✓		✓	81.67	92.51	97.13
4	✓	✓	✓	**83.48**	**95.80**	**97.76**

Table 5: **Ablation study of various similarity** with F1 metric. Text-Doc., Layout, & Text-Ent. mean textual similar documents, layout similar documents, & textual similar entities.

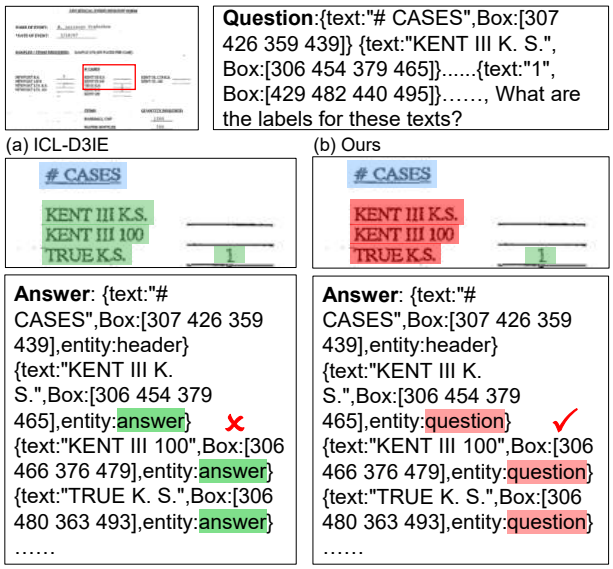

Figure 4: **Case study** on performance comparison of (a) ICL-D3IE and (b) our SAIL. ICL-D3IE wrongly predicts the three green boxes on the left as "answer". In contrast, our proposed SAIL correctly predicts them as "question".

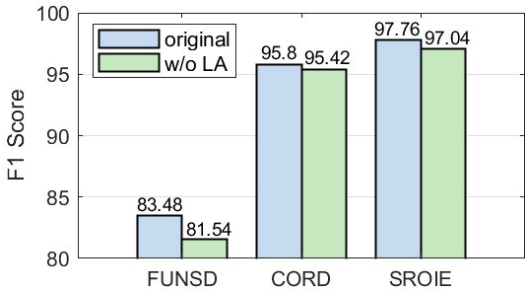

Figure 5: **Ablation study of layout analysis**. "w/o LA" means without adding layout analysis. Adding layout analysis achieves higher F1 scores across all three datasets.

effectiveness of our method. Among the three datasets, the performance improvement with adaptive examples is most pronounced in the CORD dataset (13.45%). Note that the CORD dataset contains 30 labels, much more complex than the other two datasets with only four labels. This suggests that sample-centric examples could more effectively guide the LLMs to comprehend the layout and text information especially in complex situations.

Effect of Different Examples. We conduct ablation experiments using GPT-3.5 to evaluate the influence of different examples, as shown in Table 5. In #0, where none of the three examples are available, we employ fixed random examples to instruct the LLM to generate answers in a specific format, simplifying the label extraction. The highest F1 score is observed when document-level text similarity examples, layout examples, and entity-level text similarity examples (#4) are used, validating the efficiency of the three examples. The addition of layout examples (#1 *vs.* #2) or entity-level text similarity examples (#1 *vs.* #3) to document-level text similarity examples results in superior F1 scores.

For the long text FUNSD dataset, the F1 score with document-level text similarity examples is even lower than fixed examples (#0 *vs.* #1). This could be attributed to the inherent randomness of LLM generation, but it also signi-

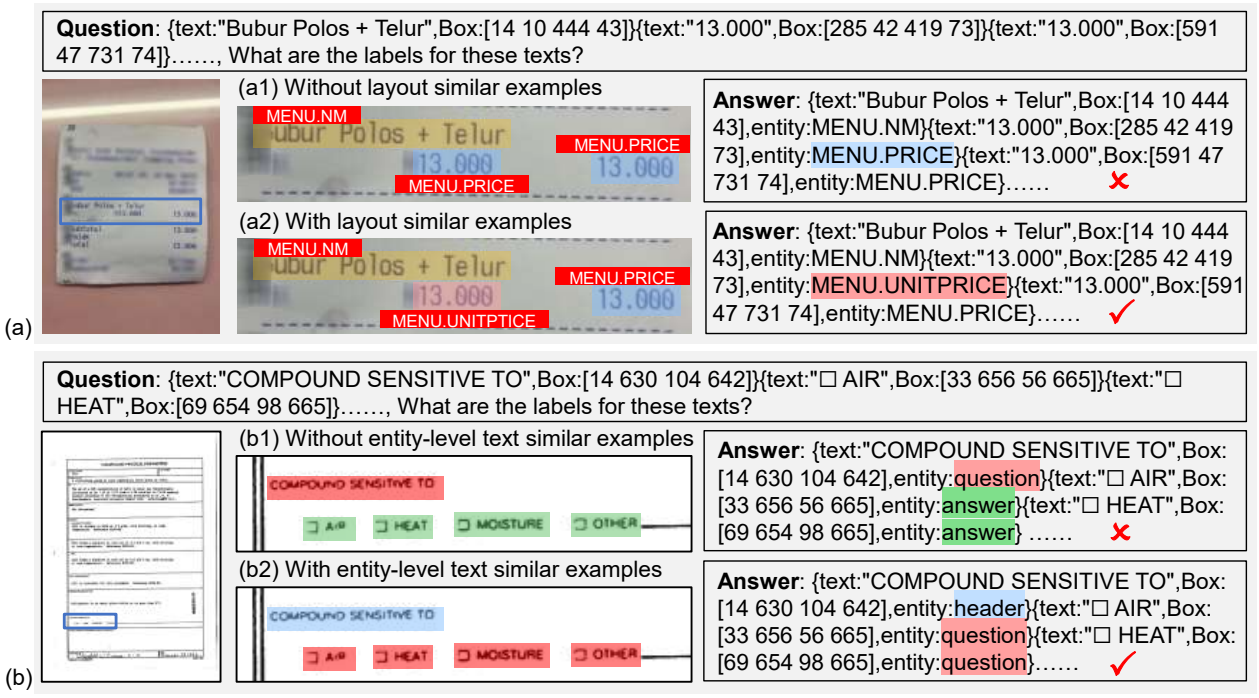

Figure 6: (a) **Case study** on comparison of (a1) without and (a2) with layout similar examples. Adding layout similar examples helps accurately distinguish between the two "13.000". (b) **Case study** on comparison of (b1) without and (b2) with entity-level text similar examples. The LLM labels entities correctly with the demonstration of entity-level text similar examples.

fies that in lengthy documents, document-level text similarity examples do not provide effective guidance for the LLM. In the FUNSD dataset, adding entity-level text similarity examples (10.35%, #2 *vs.* #4) is much superior than adding layout similarity examples (1.81%, #3 *vs.* #4), suggesting that entity-level text similarity examples are more important for lengthy documents. For the CORD and SROIE datasets, removing layout similarity examples (#3 *vs.* #4) causes a greater F1 score decrease than omitting entity-level text similarity examples (#2 *vs.* #4), indicating the higher significance of layout information for these two datasets.

Effect of Example Order. We perform experiments on the CORD dataset using GPT-3.5 to test the effect of example order, as detailed in Table 4. When layout-similar and text-similar document examples are arranged in a consistent order based on their similarity, F1 scores tend to be higher. This phenomenon may result from improved attention allocation within the LLM due to the consistent ordering. Furthermore, the highest F1 scores are observed when layout-similar and text-similar examples are sorted from high to low similarity concerning the test sample. This suggests that the LLM can capitalize on the information presented first.

Effect of Layout Analysis. Our methodology requires the LLM to perform layout analysis on our searched layout-similar examples. To assess the impact of layout analysis, we conduct comparative experiments with / without the layout analysis on the FUNSD, CORD, and SROIE datasets using the GPT-3.5. As illustrated in Figure 5, the results indicate that F1 scores are consistently higher when incorporating layout analysis compared to only using layout-similar examples across all datasets. This suggests that layout analysis is able to enhance the LLM's comprehension of layout.

Case Study. Figure 6(a) illustrates a comparison from the CORD dataset regarding the inclusion of layout demonstrations in the prompt. Using the prompt without layout similar demonstrations, the LLM predicts two "13.000" both as "MENU.PRICE", while our SAIL distinguishes the left "13.000" as "MENU.UNITPRICE" and the right "13.000" as "MENU.PRICE". This outcome underscores the necessity of incorporating layout demonstrations for LLMs to grasp document structure effectively. Figure 6(b) showcases a comparison from the FUNSD dataset about the addition of entity-level text demonstrations in the prompt. Upon omitting these demonstrations, the LLM mistakenly predicts "COMPOUND SENSITIVE TO" as "question" and incorrectly classifies the four subsequent entities as "answer". Although this prediction makes sense in terms of layout, it fails to correspond with the textual context, highlighting the critical role of entity-level text similarity examples.

5 Conclusion

In this work, we propose SAIL, a sample-centric ICL method for training-free DIE task. Our SAIL leverages layout similarity and entity-level text similarity in combination with a unified prompt template, constructing tailored prompts for each test sample, showcasing superiority over baselines on three DIE benchmarks with different LLMs.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62422311, 62176152), and Shanghai Committee of Science and Technology, China(No. 24TS1413500).

References

Abramovich, O.; Nayman, N.; Fogel, S.; Lavi, I.; Litman, R.; Tsiper, S.; Tichauer, R.; Appalaraju, S.; Mazor, S.; and Manmatha, R. 2024. VisFocus: Prompt-guided vision encoders for ocr-free dense document understanding. In *ECCV*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. In *NeurIPS*.

Cai, C.; Wang, Q.; Liang, B.; Qin, B.; Yang, M.; Wong, K.-F.; and Xu, R. 2023. In-context learning for few-shot multimodal named entity recognition. In *Findings of EMNLP*.

Da, C.; Luo, C.; Zheng, Q.; and Yao, C. 2023. Vision grid transformer for document layout analysis. In *ICCV*.

Denk, T. I.; and Reisswig, C. 2019. BERTgrid: Contextualized embedding for 2D document representation and understanding. In *NeurIPS Workshop*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Fujitake, M. 2024. LayoutLLM: Large language model instruction tuning for visually rich document understanding. In *LREC-COLING*.

Gu, Z.; Meng, C.; Wang, K.; Lan, J.; Wang, W.; Gu, M.; and Zhang, L. 2022. XYLayoutLM: Towards layout-aware multimodal networks for visually-rich document understanding. In *CVPR*.

He, J.; Wang, L.; Hu, Y.; Liu, N.; Liu, H.; Xu, X.; and Shen, H. T. 2023. ICL-D3IE: In-context learning with diverse demonstrations updating for document information extraction. In *ICCV*.

Hong, T.; Kim, D.; Ji, M.; Hwang, W.; Nam, D.; and Park, S. 2022. BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *AAAI*.

Hu, A.; Xu, H.; Ye, J.; Yan, M.; Zhang, L.; Zhang, B.; Li, C.; Zhang, J.; Jin, Q.; Huang, F.; et al. 2024. mPLUG-DocOwl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*.

Hu, Y.; Lee, C.-H.; Xie, T.; Yu, T.; Smith, N. A.; and Ostendorf, M. 2022. In-context learning for few-shot dialogue state tracking. In *Findings of EMNLP*.

Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; and Wei, F. 2022. LayoutLMv3: Pre-training for document ai with unified text and image masking. In *ACM MM*.

Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; and Jawahar, C. 2019. ICDAR2019 competition on scanned receipt ocr and information extraction. In *ICDAR*.

Jaume, G.; Ekenel, H. K.; and Thiran, J.-P. 2019. FUNSD: A dataset for form understanding in noisy scanned documents. In *ICDARW*.

Katti, A. R.; Reisswig, C.; Guder, C.; Brarda, S.; Bickel, S.; Höhne, J.; and Faddoul, J. B. 2018. Chargrid: Towards understanding 2d documents. In *EMNLP*.

Kerroumi, M.; Sayem, O.; and Shabou, A. 2021. VisualWordGrid: Information extraction from scanned documents using a multimodal approach. In *ICDAR*.

Kim, G.; Hong, T.; Yim, M.; Nam, J.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; and Park, S. 2022. Ocr-free document understanding transformer. In *ECCV*.

Li, C.; Bi, B.; Yan, M.; Wang, W.; Huang, S.; Huang, F.; and Si, L. 2021. StructuralLM: Structural pre-training for form understanding. In *ACL-IJCNLP*.

Li, X.; Wu, Y.; Jiang, X.; Guo, Z.; Gong, M.; Cao, H.; Liu, Y.; Jiang, D.; and Sun, X. 2024. Enhancing visual document understanding with contrastive learning in large visual-language models. In *CVPR*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *CVPR*.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, W. B.; Carin, L.; and Chen, W. 2022. What makes good in-context examples for GPT-3? In *DeeLIO: The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*.

Liu, W.; Lei, F.; Luo, T.; Lei, J.; He, S.; Zhao, J.; and Liu, K. 2023. MMHQA-ICL: Multimodal in-context learning for hybrid question answering over text, tables and images. *arXiv preprint arXiv:2309.04790*.

Liu, X.; Gao, F.; Zhang, Q.; and Zhao, H. 2019. Graph convolution for multimodal information extraction from visually rich documents. In *NAACL-HLT*.

Liu, Y.; Yang, B.; Liu, Q.; Li, Z.; Ma, Z.; Zhang, S.; and Bai, X. 2024b. TextMonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*.

Lu, J.; Yu, H.; Wang, Y.; Ye, Y.; Tang, J.; Yang, Z.; Wu, B.; Liu, Q.; Feng, H.; Wang, H.; et al. 2024. A bounding box is worth one token: Interleaving layout and text in a large language model for document understanding. *arXiv preprint arXiv:2407.01976*.

Luo, C.; Shen, Y.; Zhu, Z.; Zheng, Q.; Yu, Z.; and Yao, C. 2024. LayoutLLM: Layout instruction tuning with large language models for document understanding. In *CVPR*.

Mao, Z.; Bai, H.; Hou, L.; Wei, J.; Jiang, X.; Liu, Q.; and Wong, K.-F. 2024. Visually guided generative text-layout pre-training for document intelligence. In *NAACL-HLT*.

Meade, N.; Gella, S.; Hazarika, D.; Gupta, P.; Jin, D.; Reddy, S.; Liu, Y.; and Hakkani-Tür, D. 2023. Using in-context learning to improve dialogue safety. In *Findings of EMNLP*.

OpenAI. 2023a. gpt-3.5-turbo.

OpenAI. 2023b. GPT-4o System Card.

Park, S.; Shin, S.; Lee, B.; Lee, J.; Surh, J.; Seo, M.; and Lee, H. 2019. CORD: A consolidated receipt dataset for post-ocr parsing. In *NeurIPS Workshop*.

Perot, V.; Kang, K.; Luisier, F.; Su, G.; Sun, X.; Boppana, R. S.; Wang, Z.; Mu, J.; Zhang, H.; and Hua, N. 2023. LMDX: Language model-based document information extraction and localization. *arXiv preprint arXiv:2309.10952*.

Qian, Y.; Santus, E.; Jin, Z.; Guo, J.; and Barzilay, R. 2019. GraphIE: A graph-based framework for information extraction. In *NAACL-HLT*.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP*.

Tang, G.; Xie, L.; Jin, L.; Wang, J.; Chen, J.; Xu, Z.; Wang, Q.; Wu, Y.; and Li, H. 2021. MatchVIE: Exploiting match relevancy between entities for visual information extraction. *arXiv preprint arXiv:2106.12940*.

THUDM. 2023. Chatglm3.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

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

Wang, J.; Jin, L.; and Ding, K. 2022. LiLT: A simple yet effective language-independent layout transformer for structured document understanding. In *ACL*.

Wang, J.; Le, X.; Peng, X.; and Chen, C. 2023b. Adaptive hinge balance loss for document-level relation extraction. In *Findings of EMNLP*.

Wang, J.; Liu, C.; Jin, L.; Tang, G.; Zhang, J.; Zhang, S.; Wang, Q.; Wu, Y.; and Cai, M. 2021. Towards robust visual information extraction in real world: new dataset and novel solution. In *AAAI*.

Wang, L.; Hu, Y.; He, J.; Xu, X.; Liu, N.; Liu, H.; and Shen, H. T. 2024. T-SciQ: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *AAAI*.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; et al. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *ACL-IJCNLP*.

Yang, Z.; Gan, Z.; Wang, J.; Hu, X.; Lu, Y.; Liu, Z.; and Wang, L. 2022. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*.

Zhao, X.; Niu, E.; Wu, Z.; and Wang, X. 2019. CUTIE: Learning to understand documents with convolutional universal text information extractor. *arXiv preprint arXiv:1903.12363*.