ICL-D3IE: In-Context Learning with Diverse Demonstrations Updating for Document Information Extraction

Jiabang He[1], Lei Wang[2][*], Yi Hu[1], Ning Liu[3], Hui Liu[4], Xing Xu[1][†], Heng Tao Shen[1]
[1] Center for Future Media & School of Computer Science and Engineering,
University of Electronic Science and Technology of China
[2] Singapore Management University , [3] Beijing Forestry University
[4] Beijing Rongda Technology Co., Ltd.

Abstract

Large language models (LLMs), such as GPT-3 and ChatGPT, have demonstrated remarkable results in various natural language processing (NLP) tasks with in-context learning, which involves inference based on a few demonstration examples. Despite their successes in NLP tasks, no investigation has been conducted to assess the ability of LLMs to perform document information extraction (DIE) using in-context learning. Applying LLMs to DIE poses two challenges: the modality and task gap. To this end, we propose a simple but effective in-context learning framework called ***ICL-D3IE****, which enables LLMs to perform DIE with different types of demonstration examples. Specifically, we extract the most difficult and distinct segments from hard training documents as hard demonstrations for benefiting all test instances. We design demonstrations describing relationships that enable LLMs to understand positional relationships. We introduce formatting demonstrations for easy answer extraction. Additionally, the framework improves diverse demonstrations by updating them iteratively. Our experiments on three widely used benchmark datasets demonstrate that the ICL-D3IE framework enables Davinci-003/ChatGPT to achieve superior performance when compared to previous pre-trained methods fine-tuned with full training in both the in-distribution (ID) setting and in the out-of-distribution (OOD) setting. Code is available at* `https://github.com/MAEHCM/ICL-D3IE`.

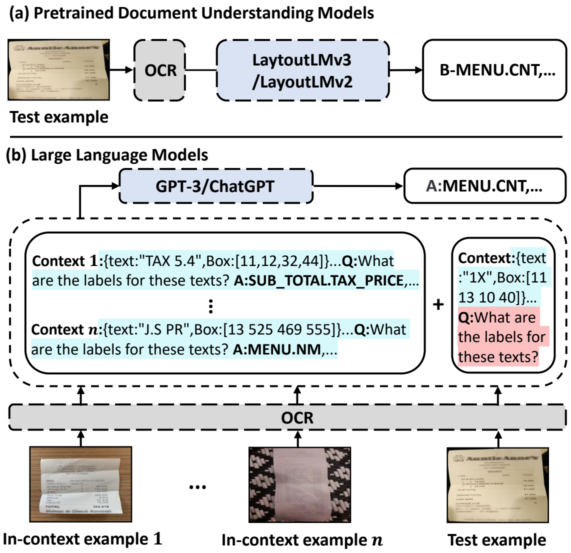

(a) Pretrained Document Understanding Models
OCR
LaytoutLMv3 /LayoutLMv2
B-MENU.CNT,...
Test example
(b) Large Language Models
GPT-3/ChatGPT
A:MENU.CNT,...
Context 1:{text:"TAX 5.4",Box:[11,12,32,44]}...Q:What are the labels for these texts? A:SUB_TOTAL.TAX_PRICE,...
⋮
Context n:{text:"J.S PR",Box:[13 525 469 555]}...Q:What are the labels for these texts? A:MENU.NM,...
+
Context:{text:"1X",Box:[11 13 10 40]}... Q:What are the labels for these texts?
OCR
...
In-context example 1
In-context example n
Test example

Figure 1: Two approaches for solving the DIE task: (a) previous pre-trained document understanding models [15, 42] fine-tuned with full training examples, and (b) in-context learning over LLMs with a few examples.

1. Introduction

The task of visually rich document understanding (VRDU), which involves extracting information from VRDs [2, 19], requires models that can handle various types of documents, such as voice, receipts, forms, emails, and advertisements, and various types of information, including rich visuals, large amounts of text, and complex document layouts [28, 18, 26]. Recently, fine-tuning based on pre-trained visual document understanding models has yielded impressive results in extracting information from VRDs [41, 13, 22, 23, 15, 21], suggesting that the use of large-scale, unlabeled training documents in pre-training document understanding models can benefit information extraction from VRDs. As shown in Figure 1 (a), a pretrained model such as LayoutLMv3 [15] can predict labels for entities in a test VRD.

Large language models (LLMs), such as GPT-3 [1],

[*] Corresponding author. lei.wang.2019@phdcs.smu.edu.sg
[†] Corresponding author. xing.xu@uestc.edu.cn

OPT [45], and PaLM [5], develop quickly and have shown remarkable results in various natural language processing (NLP) tasks. As LLMs grow in model parameters and training corpus size, they reveal emergent abilities that allow them to learn to reason from just a few demonstration examples within a given context [38]. This paradigm of learning is referred to as in-context learning (ICL) [8]. Recently, approaches [43, 14] have been proposed to explore how to use LLMs to solve vision-language (VL) tasks. However, to date, there has been no investigation into the ability of LLMs to handle VRD understanding tasks, such as document information extraction (DIE). Similar to VQA [12], Two main challenges arise when applying LLMs to DIE: the modality gap and the task gap, as LLMs cannot directly process images and may lack training on layout information in VRDs.

To address these challenges, one popular strategy in using LLMs for the VQA task is to use demonstration QA pairs and convert their corresponding images into image descriptions through image caption models [43, 14]. Subsequently, the demonstration QA pairs and image descriptions are combined as a prompt for the LLM to answer a test question. Figure 1 (b) shows this straightforward strategy to apply LLMs to the DIE task. It first utilizes Optical Character Recognition (OCR) tools to convert images of demonstration documents from the training data into textual contents and corresponding entity bounding boxes. The converted demonstrations with entity labels are then combined as a prompt for LLMs to predict labels for entities in a test document. However, this strategy may perform poorly, as it ignores positional relationships among textual contents and is sensitive to examples selected for demonstrations.

In this paper, we propose **ICL-D3IE**, a simple and effective in-context learning framework for LLMs to perform the DIE task with various types of demonstration examples within a given context. Our method constructs different types of demonstrations based on three criteria: (1) the demonstrations should benefit all test documents rather than just a subset of them, (2) layout information must be included, and (3) the demonstrations should predict labels in an easily extractable format. To construct hard demonstrations for the first criterion, we select challenging segments from the training documents that are difficult for LLMs to accurately predict entities. To construct layout-aware demonstrations for the second criterion, we use a prompt question to direct LLMs to describe positional relationships between textual content boxes in selected regions. To create formatting demonstrations for the third criterion, we randomly choose training segments to guide LLMs to predict labels in a desired format for easy extraction. Furthermore, the framework iteratively enhances diverse demonstrations by updating hard demonstrations through in-context learning with previous diverse demonstrations.

Experiments conducted on three widely used benchmark datasets (FUNSD [18], CORD [28], and SROIE [16]), demonstrate that ICL-D3IE allows LLMs to achieve DIE performance that is superior or comparable to previous pre-trained methods fine-tuned with full training samples when tested in the in-distribution (ID) setting. For example, ICL-D3IE with GPT-3 (97.88%) outperforms LayoutLMv3$_{\mathrm{base}}$ (96.89%) on SROIE. Moreover, in the out-of-distribution (OOD) setting, ICL-D3IE performs much better than previous pre-trained methods on all datasets, achieving superior performance. Together, these remarkable results encourage new ways to leverage LLMs for solving VRD-related tasks.

2. Related Work

Visually Rich Document Understanding (VRDU). The research topic of VRDU has been a challenging area of research for many years, with numerous named entity recognition (NER) methods proposed based on neural networks, such as recurrent neural networks [20]. However, most of these methods only identify key information in plain text, neglecting the visual and layout information present in the document. To address this issue, convolutional and graph neural networks have been introduced to model layout and semantic information [46, 24]. Recently, multimodal self-supervised pre-training and fine-tuning have proven effective in visually rich documents by modeling visual, layout, and textual information [40, 35, 11, 36, 15]. Huang et al. [15] were inspired by the Vision Transformer (ViT) [10] to use patch-level embeddings to learn visual features in LayoutLMv3. DIE involves automatically extracting information from VRDs. The objective is to identify valuable information in these complex documents and organize it in a format that can be easily analyzed and used. The process of extracting information from VRDs requires two essential steps: (1) text detection and recognition in document images, and (2) entity labeling of the recognized text. The first step falls under the area of research known as optical character recognition. This study focuses on the second step and mainly discusses how to leverage GPT-3 to accurately label entities in recognized text.

In-Context Learning. LLMs like GPT-3 [1], OPT [45], and PaLM [5] demonstrate emergent abilities as model and corpus sizes increase [38]. These abilities are learned from demonstrations containing a few examples in the context, which is known as in-context learning [8]. To enable reasoning in LLMs, [39] propose Chain-of-Thought (CoT) prompting, which adds multiple reasoning steps to the input question. CoT prompting is a simple and effective few-shot prompting strategy that improves LLMs' performance on complex reasoning tasks. Several works [34, 32, 31] have since aimed to improve CoT prompting in different aspects, such as prompt format [4], prompt selection [25], prompt ensemble [37], and problem decomposition [47].

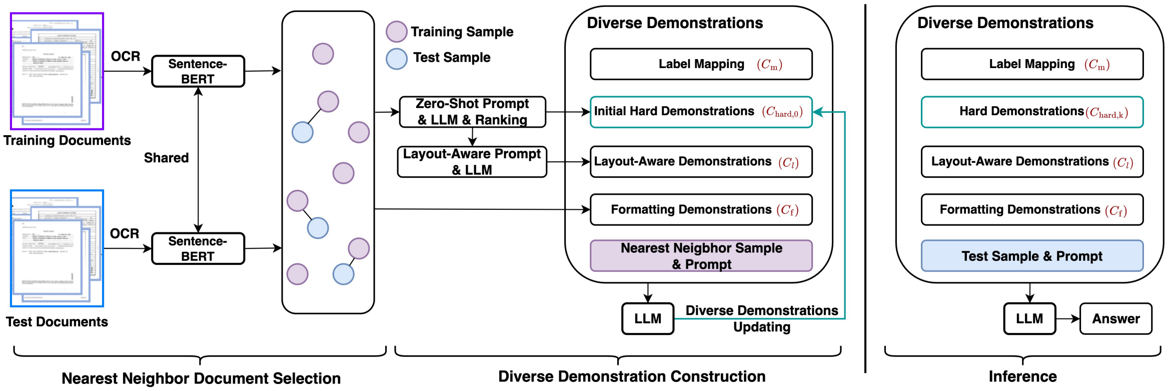

OCR
Sentence-BERT
Training Documents
Shared
OCR
Sentence-BERT
Test Documents
Training Sample
Test Sample
Zero-Shot Prompt & LLM & Ranking
Layout-Aware Prompt & LLM
Diverse Demonstrations
Label Mapping (C_{m})
Initial Hard Demonstrations $(C_{\mathrm{hard},0})$
Layout-Aware Demonstrations (C_l)
Formatting Demonstrations (C_{f})
Nearest Neigbhor Sample & Prompt
LLM
Diverse Demonstrations Updating
Diverse Demonstrations
Label Mapping (C_{m})
Hard Demonstrations $(C_{\mathrm{hard,k}})$
Layout-Aware Demonstrations (C_l)
Formatting Demonstrations (C_{f})
Test Sample & Prompt
LLM
Answer
Nearest Neighbor Document Selection
Diverse Demonstration Construction
Inference

Figure 2: A detailed illustration of ICL-D3IE framework, including obtaining nearest neighbor documents for test samples from the training dataset, constructing iteratively updated diverse demonstrations, and performing inference.

While LLMs were originally developed for NLP tasks, recent studies[43, 3, 44] have shown that LLMs with in-context learning have few-shot or zero-shot abilities for multimodal problems, including visual question answering tasks. Furthermore, Frozen [33] demonstrates promising few-shot performance using pre-trained models for vision-and-language tasks. However, to our knowledge, our work is the first to explore the use of LLMs with in-context learning for information extraction from VRDs. You can refer to [9] for more related works on in-context learning

3. Our ICL-D3IE Method

3.1. Preliminary of In-Context Learning

In-context learning enables LLMs to quickly adapt to solve downstream tasks using just a few examples during inference [1], requiring no training. In contrast, fine-tuning LLMs necessitates training on as many samples as feasible, resulting in redundant computation and time expenses. This section describes how to formulate in-context learning for solving the DIE task.

A data sample consists of a document image I and its corresponding entity labels $Y = \{y_1, y_2, ..., y_L\}$, where L is the number of entities in the document. To obtain textual contents and their corresponding boxes, we process the document image I using an OCR tool. We denote the set of textual contents as $T = \{t_1, t_2, ..., t_L\}$, where t_l is a segment of words, and denote the set of their corresponding boxes as $B = \{b_1, b_2, ..., b_L\}$, where b_l is the coordinates $\left[p_1^l, p_2^l, p_3^l, p_4^l\right] \in \mathbb{Z}^4$ of the box b_l. Note that the ordering of T is crucial because GPT-3 is sensitive to the permutation of words. We follow the approach of XYLayoutLM [11] and use the XYCut algorithm to determine the ordering of textual regions. The DIE task (This paper considers the task of entity labeling in VRDs) involves generating labels Y for the given entities T in the document image I by maximizing the conditional probability as follows: $p(Y \mid T) = \frac{1}{L} \sum_l^L p\left(y_l \mid t_l\right)$.

While previous state-of-the-art studies [40, 11] typically fine-tune pre-trained models to downstream tasks, this paper proposes using LLMs with in-context learning to solve the DIE task. Specifically, we define the probability of generating the target entity labels Y for a given document image $\mathbf{I}$ and in-context string C using a LLM $\mathcal{P}_{lm}$ as follows:

$$p(Y|I, C) = \sum_{l=1}^{L} \mathcal{P}_{lm}\left(\mathcal{V}(y_l)|C, \mathcal{T}(I)\right). \qquad (1)$$

Here, $\mathcal{T}(\cdot)$ denotes a set of operations used to convert the original document image into a text format as GPT-3 desire, C is the in-context examples obtained by concatenating k input-output demonstration examples $\{(\mathcal{T}(I_1), Y_1), (\mathcal{T}(I_2), Y_2), \ldots, (\mathcal{T}(I_k), Y_k)\}$, and $\mathcal{V}$ is an operation for mapping an entity label y_l to natural language words that can be understood by GPT-3.

3.2. Overview Framework of ICL-D3IE

We present ICL-D3IE, a novel in-context learning framework for tackling the DIE task, that enables GPT-3 to predict entity labels in a test document based on different types of demonstrations. Constructing demonstrations is designed to satisfy three criteria: (i) the demonstrations should benefit all test documents, not just a subset, (ii) they should include layout information, which is essential for solving VRD-related tasks., and (iii) they should predict entity labels in an easily extracted and evaluated format.

The proposed ICL-D3IE framework involves four key steps as shown in Figure 2. Firstly, the framework selects n training documents most similar to the n test documents. Secondly, ICL-D3IE constructs diverse demonstrations based on the selected similar training documents. These demonstrations include initial hard demonstrations for criterion (i), layout-aware demonstrations for criterion (ii), and formatting demonstrations for criterion

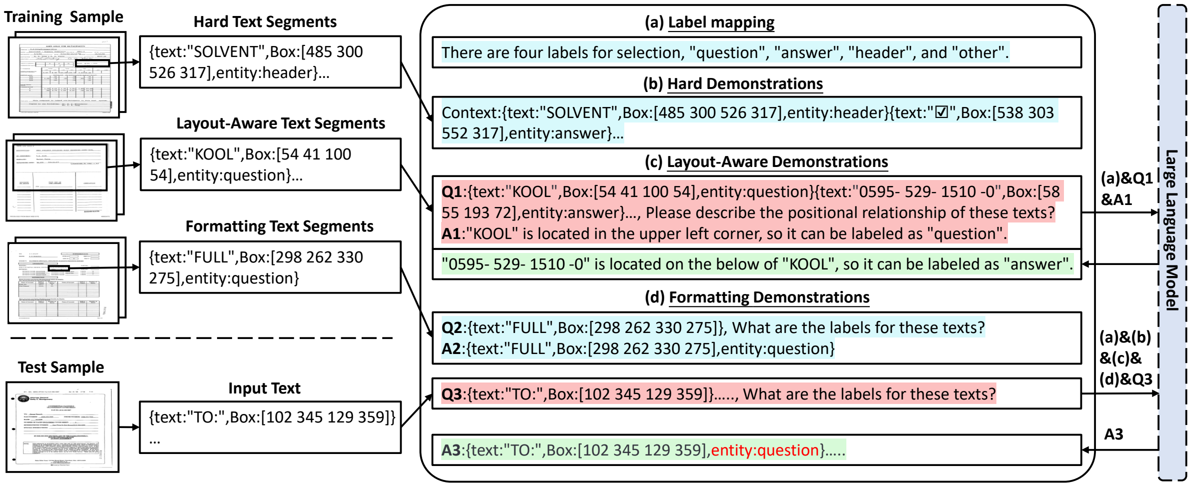

Training Sample
Hard Text Segments
{text:"SOLVENT",Box:[485 300 526 317],entity:header}...
Layout-Aware Text Segments
{text:"KOOL",Box:[54 41 100 54],entity:question}...
Formatting Text Segments
{text:"FULL",Box:[298 262 330 275],entity:question}
Test Sample
Input Text
{text:"TO:",Box:[102 345 129 359]}
...
(a) Label mapping
There are four labels for selection, "question", "answer", "header", and "other".
(b) Hard Demonstrations
Context:{text:"SOLVENT",Box:[485 300 526 317],entity:header}{text:"☑",Box:[538 303 552 317],entity:answer}...
(c) Layout-Aware Demonstrations
Q1:{text:"KOOL",Box:[54 41 100 54],entity:question}{text:"0595- 529- 1510 -0",Box:[58 55 193 72],entity:answer}..., Please describe the positional relationship of these texts?
A1:"KOOL" is located in the upper left corner, so it can be labeled as "question".
"0595- 529- 1510 -0" is located on the below of "KOOL", so it can be labeled as "answer".
(d) Formatting Demonstrations
Q2:{text:"FULL",Box:[298 262 330 275]}, What are the labels for these texts?
A2:{text:"FULL",Box:[298 262 330 275],entity:question}
Q3:{text:"TO:",Box:[102 345 129 359]}....., What are the labels for these texts?
A3:{text:"TO:",Box:[102 345 129 359],entity:question}.....
(a)&Q1
&A1
(a)&(b)
&(c)&
(d)&Q3
A3
Large Language Model

Figure 3: Example of the input and output of in-context learning with diverse demonstrations. The text highlighted in blue is not processed by LLMs, while the text highlighted in red is fed into LLMs. The green-highlighted text represents the output of LLMs. The text in red represents the prediction made by the LLM. The final prompt comprises label mapping, hard demonstrations, layout-aware demonstrations, formatting demonstrations, and a question prompt of "What are the labels for these texts?".

(iii). Thirdly, the framework iteratively updates the diverse demonstrations by improving the hard demonstrations through in-context learning with previous diverse demonstrations. Lastly, ICL-D3IE performs inference using in-context learning with the updated diverse demonstrations.

3.3. Nearest Neighbor Document Selection

To facilitate effective in-context learning, the proposed ICL-D3IE selects n training documents that are most similar to the n test documents. This process involves several steps. Firstly, we leverage OCR tools to convert m training and n test document images into plain text with corresponding box information. Subsequently, the plain text is fed into Sentence-BERT [30] to obtain document representations, and cosine similarity scores are calculated to identify the most similar training document for each test document. Finally, we can identify n training documents that are the closest match to the n test documents, which we refer to as nearest neighbor documents $I_1^{\text{nnd}}, I_2^{\text{nnd}}, \ldots, I_n^{\text{nnd}}$.

3.4. Diverse Demonstrations Construction

Once we have obtained n nearest neighbor documents from the training dataset, we construct diverse demonstrations for effective in-context learning. The standard approach to constructing in-context demonstrations involves designing a template for the target task to convert data examples into texts that LLMs can process. Unlike standard in-context learning that relies solely on task-specific demonstrations, ICL-D3IE constructs diverse demonstrations for each test instance: hard demonstrations that highlight challenging aspects of a task, layout-aware demonstrations that describe the positional relationship between textual contents, and formatting demonstrations that provide output formatting examples.

Initial Hard Demonstrations. The first criterion for selecting hard demonstrations is that they should highlight the most challenging aspects of the DIE task to benefit all test documents. The process of obtaining initial hard demonstrations involves several steps. First, we use a zero-shot prompting technique, which involves using a prompt such as "What are the labels for these texts?" pt_0 to ask GPT-3 to predict labels for entities in I_i^{nnd}. Next, we calculate entity-level F1 scores based on the predicted labels and the corresponding ground truth labels. We then identify the text segment t_{hard} with the lowest F1 scores from the nearest neighbor documents. An initial hard demonstration can be formulated as:

$$C_{\text{hard},0} = \text{CONCAT}(t_{\text{hard}}, b_{\text{hard}}, pt_0, y_{\text{hard}}), \quad (2)$$

where b_{hard} and y_{hard} are the box coordinate and answer of the text segment t_{hard}, respectively.

Layout-Aware Demonstrations. The second criterion necessitates the inclusion of layout information in the in-context demonstrations, which is crucial for completing the DIE task. To acquire demonstrations mindful of layout, We randomly select adjacent hard segments obtained in the construction of $C_{\text{hard},0}$ to create a region R_l for positional relationship description. We use a prompt "Please describe the positional relationship of these texts" pt_l to guide GPT-3 to generate a description $\tilde{y}_l$ of the positional relationship between text segments in R_l. A layout-aware demonstration can be formulated as:

$$C_l = \mathrm{CONCAT}(R_l, B_l, pt_l, \tilde{y}_l), \tag{3}$$

where B_l are the box coordinates for text segments of the selected region R_l.

Formatting Demonstrations. The third criterion expects to provide examples to guide GPT-3 to format the output for the DIE task. To achieve this, we first randomly select a text segment t_f from the nearest neighbor documents. Then, a formatting demonstration C_f consist of a text segment t_f, its corresponding box coordinate b_f, the formatting prompt pt_0, and the ground truth answer y_f, denoted as C_f:

$$C_\mathrm{f} = \mathrm{CONCAT}(t_\mathrm{f}, b_\mathrm{f}, pt_0, y_\mathrm{f}). \tag{4}$$

Label Mapping. The objective of label mapping is to translate unnatural word labels to an answer space where GPT-3 can effectively function as a predictive model. To achieve this, we gather text descriptions of the original labels from the provided datasets, such as "total. cash price" representing "the amount paid in cash." Then, we include the original labels (Y') and their corresponding descriptions (Y) in the context before various demonstrations to prompt GPT-3 to solve the test sample. Label Mapping for prompting can be formulated as:

$$C_\mathrm{m} = \mathrm{CONCAT}(Y', Y). \tag{5}$$

3.5. Diverse Demonstrations Updating

To further highlight the most challenging aspects of the DIE task, ICL-D3IE iteratively updates its diverse demonstrations by improving hard demonstrations through in-context learning with previous diverse demonstrations. Initial diverse demonstrations with initial hard demonstrations $C_{\mathrm{hard},0}$ are used to perform inference for all nearest neighbor documents $I_1^{\mathrm{nnd}}, I_2^{\mathrm{nnd}}, \ldots, I_n^{\mathrm{nnd}}$. Entity-level F1 scores are computed for all entities, and the text segment with the lowest F1 score is appended to the initial hard demonstrations to obtain new hard demonstrations $C_{\mathrm{hard},1}$. This process is iterated k times to obtain final updated hard demonstrations $C_{\mathrm{hard,k}}$, which are used to construct the final diverse demonstrations.

3.6. Inference

After diverse demonstrations updating, the obtained diverse and comprehensive demonstrations can be used to direct GPT-3 to perform the test, which is formulated as follows:

$$p(Y|\mathbf{I}, C) = \frac{1}{L}\sum_{l=1}^{L} \mathcal{P}_{lm}\left(\mathcal{V}(y_l)|C_\mathrm{m}, C_{\mathrm{hard,k}}, C_l, C_\mathrm{f}, \mathcal{T}(\mathbf{I})\right). \tag{6}$$

Finally, ICL-D3IE extracts the corresponding answers from the generated predictions and then converts them into a suitable format for evaluation.

4. Experiment

4.1. Experimental Setup

Datasets. We experiment on three widely used DIE datasets. Here is a brief introduction to these datasets: The `FUNSD` dataset [17] is a noisy scanned form understanding dataset. It comprises 199 documents with varying layouts and 9,707 semantic entity annotations in total. In our study, we focus on the semantic entity labeling task, which involves assigning labels such as "question," "answer," "header," or "other" to each semantic entity. The training set comprises 149 samples, and the test set comprises 50 samples. The `CORD` dataset [29] is a consolidated receipt understanding dataset that includes 800 receipts for training, 100 receipts for validation, and 100 receipts for testing. The labels in this dataset have a hierarchy, comprising 30 semantic labels under four categories. However, the labels are more complex than those in the FUNSD dataset and require label mapping. The `SROIE` dataset [16] is another receipt understanding dataset, comprising 973 receipts categorized into four classes. The dataset includes 626 training images and 347 test images. The labels in this dataset are "company," "date," "address," and "total."

Baselines. We compare ICL-D3IE with three types of baselines. The first type includes strong pre-trained models fine-tuned with full training samples, while the second type includes those fine-tuned with only a few samples. The third type includes standard in-context learning, where one of its demonstrations includes one document's textual contents, the corresponding box coordinates, the prompt question pt_0, and the corresponding ground truth answers.

For the text modality-based pre-trained baseline, we compare our method to BERT [6]. For the text and layout modalities based on pre-trained baselines, we employ LiLT [35] and BROS [13]. LiLT uses a language-independent layout transformer that decouples text and layout modalities. BROS is a pre-trained key information extraction model that encodes relative layout information. Furthermore, we also consider pre-trained baselines that utilize text, layout, and image modalities, including LayoutLM [40], XYLayoutLM [11], LayoutLMv2 [42], and LayoutLMv3 [15]. LayoutLM uses two objectives to learn language representation during pre-training and incorporates image information during the fine-tuning phase. XYLayoutLM employs a preprocessing algorithm called Augmented XY Cut to generate proper reading orders. LayoutLMv2 uses CNN to encode document images and utilizes image information during the pre-training stage. Lastly, LayoutLMv3 can model patch-level document information.

Implementation Details. In our experiments, we use the public GPT-3 `text-davinci-003` (175B) and Chat-

	Dataset	FUNSD			CORD			SROIE		
Setting	Model	ID	OOD	Average	ID	OOD	Average	ID	OOD	Average
		F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑
Full-Training	BERT$_{\text{BASE}}$ [7]	60.26	51.02	55.64	89.68	55.68	72.68	90.99	72.36	81.68
	LiLT$_{\text{BASE}}$ [35]	88.41	64.29	76.35	96.07	73.32	84.70	94.68	74.29	84.49
	BROS$_{\text{BASE}}$ [13]	83.05	68.72	75.89	95.73	71.24	83.49	95.48	75.51	85.50
	XYLayoutLM$_{\text{BASE}}$ [11]	83.35	61.24	72.30	94.45	69.12	81.79	95.74	75.91	85.83
	LayoutLM$_{\text{BASE}}$ [41]	79.27	54.38	66.83	91.06	70.13	80.60	94.38	76.24	85.31
	LayoutLMv2$_{\text{BASE}}$ [42]	82.76	59.66	71.21	94.95	76.39	85.67	96.25	78.57	87.41
	LayoutLMv3$_{\text{BASE}}$ [15]	90.29	73.24	81.77	**96.56**	75.23	85.90	96.89	78.34	87.62
Few-Shot	BERT$_{\text{BASE}}$ [7]	38.76	19.68	29.22	38.88	15.31	27.10	38.76	20.56	59.32
	LiLT$_{\text{BASE}}$ [35]	54.88	25.32	40.10	69.12	29.94	49.53	84.03	61.25	72.64
	BROS$_{\text{BASE}}$ [13]	59.46	27.49	43.48	72.78	36.34	54.56	76.78	57.28	67.03
	XYLayoutLM$_{\text{BASE}}$ [11]	65.44	30.56	48.00	69.16	32.19	50.68	75.66	56.23	65.95
	LayoutLM$_{\text{BASE}}$ [41]	32.49	17.66	25.08	40.19	23.62	31.91	76.79	55.44	66.12
	LayoutLMv2$_{\text{BASE}}$ [42]	71.42	49.12	60.27	65.71	29.43	47.57	81.81	59.56	70.69
	LayoutLMv3$_{\text{BASE}}$ [15]	70.67	48.33	59.50	70.13	32.88	51.51	79.13	56.08	67.61
	Standard ICL (ChatGPT)	72.76	69.32	71.04	68.34	65.68	67.01	82.11	80.31	81.21
	Standard ICL (Davinci-003)	71.52	67.31	69.42	67.96	64.28	66.12	79.34	76.12	77.73
	ICL-D3IE (ChatGPT)	83.66	79.05	81.36	87.13	70.69	78.91	92.63	86.31	89.47
	ICL-D3IE (Davinci-003)	**90.32**	**88.71**	**89.52**	94.12	**91.23**	**92.68**	**97.88**	**93.76**	**95.82**

Table 1: Results of comparing ICL-D3IE with Standard ICL and existing pre-trained VDU models fine-tuned with full training samples and a few samples on three benchmark datasets in ID and OOD settings.

GPT `gpt-3.5-turbo` with the API[1] as the backbone language models due to their popularity and accessibility. To ensure consistent output, we set the temperature parameter to 0. For evaluation, we employ the same metrics as in LayoutLMv3 and reported entity-level F1 for all methods. For our ICL-D3IE method, we use 4 hard demonstrations, 4 layout-ware demonstrations, and 4 formatting demonstrations. For the fine-tuning-based baselines, we adopt the hyper-parameters reported in their original papers. Note that our demonstrations may be segments, and we use document examples that include segments used in our method to fine-tune few-shot baseline models to ensure a fair comparison. To demonstrate the generalization ability of in-context learning over LLMs, we generate out-of-distribution (OOD) test data for three benchmark datasets using TextAttack [27]. The original test data for these datasets are referred to as in-distribution (ID) test data. Specifically, we replace original words with words that are nearly identical in appearance yet different in meaning and delete certain characters in words, such as "name" ⟶ "nme," to generate OOD samples.

4.2. Main Results

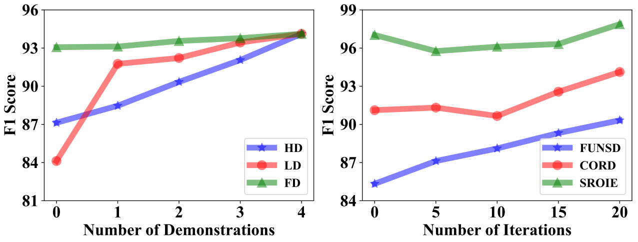
Table 1 presents the performance comparison of ICL-D3IE with existing full-training and few-shot baseline methods on both in-domain (ID) and out-of-domain (OOD) settings. We can first observe that on the ID setting, ICL-D3IE (Davinci-003) achieves a new state-of-the-art on FUNSD and SROIE datasets with only a few data examples and without any training. It achieves 90.32% on FUNSD and 97.88% on SROIE, beating all other VDU achieving SOTA. On the SROIE dataset, ICL-D3IE (Davinci-003) reaches within 3% of the state-of-the-art performance, which is comparable to pre-trained VDU models that are fine-tuned with full training samples. On the other hand, ICL-D3IE has large performance gains for DIE in the few-shot setting. For instance, in CORD, average performance more than doubled for the VDU in the few-shot setting. Meanwhile, compared to other full-training baselines, ICL-D3IE has greater robustness to OCR errors in document content on the OOD settings, resulting in significantly better performance.

Moreover, we can see that ICL-D3IE outperforms Standard ICL on three datasets, with ICL-D3IE (Davinci-003) showing an 18.8 F1 score improvement over Standard ICL (Davinci-003) on FUNSD. We experiment with GPT-3 (`text-davinci-003`) and ChatGPT (`gpt-3.5-turbo`) to investigate the applicability of ICL-D3IE with different backbone language models and find that ICL-D3IE substantially improves the performance of ChatGPT compared with Standard ICL. However, ChatGPT generation's flexibility makes answer extraction harder, re-

[1] https://platform.openai.com/docs/models/gpt-3-5

	FUNSD F1↑	CORD F1↑	SROIE F1↑
ICL-D3IE	90.32	94.12	97.88
w/o HD	78.20	87.13	89.13
w/o LD	87.25	84.13	96.83
w/o LM	89.63	87.94	97.19
w/o FD	88.73	93.07	90.58

Table 2: The effect of different components in ICL-D3IE. HD means Hard Demonstrations. LD means Layout-Aware Demonstrations. LM means Label Mapping. FD means Formatting Demonstrations.



(a)

(b)

Figure 4: Further analysis on (a) the effect of the number of different demonstrations on CORD and (b) the effect of the number of hard demonstrations updating.

sulting in slightly worse performance for ICL-D3IE (ChatGPT) compared to ICL-D3IE (Davinci-003). These promising results demonstrate the effectiveness of ICL-D3IE for the DIE task and its versatility across different backbone language models.

Overall, our ICL-D3IE method shows consistent superiority over other methods across all datasets and settings except for the ID setting on CORD, suggesting its potential to effectively solve VRD-related tasks using LLMs. These remarkable results not only highlight the effectiveness of ICL-D3IE but also inspire the development of novel methods with LLMs that require less manual effort.

4.3. Further Analysis

In this section, we conduct a detailed analysis of ICL-D3IE and its components.

Effect of Different Components in Diverse Demonstrations. ICL-D3IE's demonstrations consist of four components: hard demonstrations, layout-aware demonstrations, formatting demonstrations, and label mapping. In this section, we evaluate the impact of each component by removing one at a time and measuring the effect on ICL-D3IE (Davinci-003) performance.

As shown in Table 2, removing any components drops DIE performance. Removing hard demonstrations has the most significant impact, indicating the effectiveness of iteratively updated hard demonstrations in benefiting all test samples. Removing layout-aware demonstrations leads to a drop of around 10 F1 score on CORD but little on SROIE since CORD labels require more layout information than SROIE. Removing label mapping results in a significant drop in CORD due to its unnatural labels. ICL-D3IE's performance without label mapping suggests formatting demonstrations contribute to easier and better answer extraction. Notably, ICL-D3IE (Davinci-003) outperforms Standard ICL (Davinci-003) (Table 1), even with one component removed. Overall, these results highlight the effectiveness of each component in ICL-D3IE's in-context demonstrations.

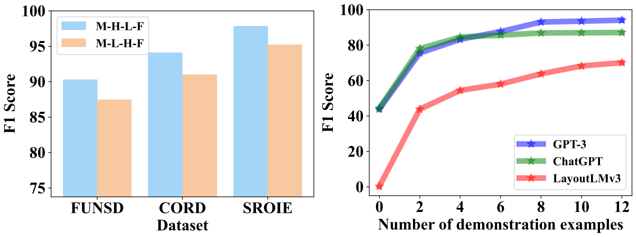
Effect of the Number of each Type of Demonstrations. In Table 1, we set the number of different types of demonstrations in ICL-D3IE to 4. However, varying the number of each type of demonstration in the in-context diverse demonstrations may result in varying performance outcomes. To investigate this, we vary the number of a specific type of demonstration from 0 to 4 while keeping the number of other types of demonstrations constant at 4.

We present the F1 score of ICL-D3IE (Davinci-003) on CORD in Figure 4a. We can observe that the number of demonstrations of each type influences the performance of ICL-D3IE. Besides, performance improves as the number of any demonstration increases. Interestingly, we observe significant changes in performance when varying the number of hard and layout-aware demonstrations, suggesting that hard demonstrations are beneficial for solving all test samples and that the DIE task on CORD requires a substantial amount of layout information to solve.

Effect of the Number of Hard Demonstrations Updating. This study aims to investigate the impact of the number of Hard Demonstrations Updating on three different datasets. As highlighted in Figure 4b, initial hard demonstrations can help ICL-D3IE work very well, and hard demonstrations after 20 iterations can achieve better performance. These findings demonstrate that incorporating feedback from challenging aspects, as identified through predictions on training data, to the prompt for LLMs is a useful strategy that can benefit solving all test samples. Additionally, updating Hard Demonstrations through in-context learning with previous diverse demonstrations can enhance the performance of ICL-D3IE (Davinci-003).

Effect of Ordering of Diverse Demonstrations. This study investigates the impact of the different ordering of demonstrations on ICL-DI3E (Davinci-003) performance. Specifically, we change the ordering of hard and layout-ware demonstrations and evaluate two different orderings: M-H-L-F (label mapping, hard demonstrations, layout-aware demonstrations, and formatting demonstrations) and M-L-H-F (label mapping, layout-aware demonstrations, hard demonstrations, and formatting demonstrations).

Figure 5a presents a comparison of the performance of



(a) (b)

Figure 5: Further analysis on (a) the performance effect of arranging demonstrations in a different order and (b) the performance comparison of increasing the number of demonstrations on ICL-D3IE (Davinci-003/ChatGPT) and LayoutLMv3 on CORD.

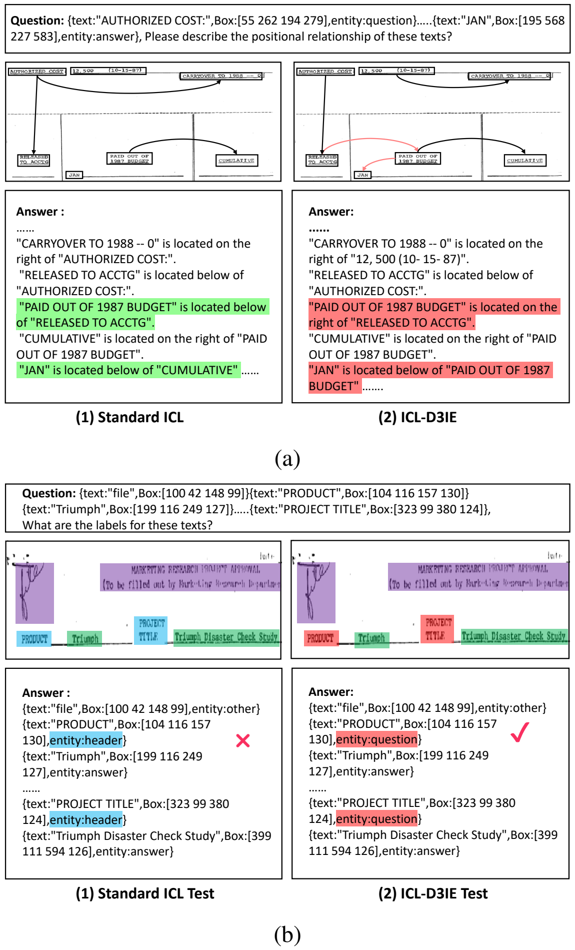
these two orderings. In our case, M-H-L-F consistently outperforms M-L-H-F across all three datasets. It suggests that in-context learning is highly sensitive to the ordering of demonstrations and that finding the optimal ordering for in-context learning is critical. Our study highlights the importance of optimizing the ordering of demonstrations for in-context learning, and this will be a focus of our future research.

Effect of the Number of Demonstration Examples. To further evaluate the performance of ICL-D3IE in comparison to pre-trained VRDU models fine-tuned with a few demonstrations, we varied the number of demonstrations for ICL-D3IE (Davinci-003), ICL-D3IE (ChatGPT), and LayoutLMv3 from 1 to 12. Figure 5b demonstrates that the performances of all three methods improve as the number of demonstrations increases on CORD. Notably, ICL-D3IE (Davinci-003) and ICL-D3IE (ChatGPT) consistently outperform LayoutLMv3 by a large margin across all numbers of demonstrations. These results suggest that our proposed in-context diverse demonstrations approach is effective and outperforms pre-trained VRDU models fine-tuned with a few demonstrations.

Case Study. Figure 6a presents two examples of asking positional relationship descriptions with Standard ICL (Davinci-003) and ICL-D3IE (Davinci-003) during the test phase. Our results illustrate that Standard ICL, without layout-ware demonstrations, cannot accurately describe the positional relationships between textual contents in a document, while ICL-D3IE can do so effectively. In Figure 6b, we observe that Standard ICL predicts the entities in the blue box as "header," while ICL-D3IE predicts the entities as "question."These findings highlight the importance of applying diverse demonstrations such as hard and layout-aware demonstrations in DIE tasks.

5. Conclusion

In this paper, we proposed ICL-D3IE, an in-context learning framework that addresses the challenges of applying LLMs to DIE tasks, specifically the modality and task gap. We extracted challenging segments from hard training documents to benefit all test instances, designed demonstrations that describe positional relationships to enable LLMs to understand the layout of documents, and introduced formatting demonstrations to facilitate easy answer extraction. The framework also improves diverse demonstrations iteratively and uses label mapping to convert unnatural words to words that GPT can process. Our evaluation of three DIE datasets shows that ICL-D3IE consistently outperforms other methods, except for the ID setting on CORD. These results highlight the potential of in-context learning frameworks for VRD understanding tasks based on LLMs, and we hope to inspire future research in this area.



Question: {text:"AUTHORIZED COST:",Box:[55 262 194 279],entity:question}.....{text:"JAN",Box:[195 568 227 583],entity:answer}, Please describe the positional relationship of these texts?

Answer :
......
"CARRYOVER TO 1988 -- 0" is located on the right of "AUTHORIZED COST:".
"RELEASED TO ACCTG" is located below of "AUTHORIZED COST:".
"PAID OUT OF 1987 BUDGET" is located below of "RELEASED TO ACCTG".
"CUMULATIVE" is located on the right of "PAID OUT OF 1987 BUDGET".
"JAN" is located below of "CUMULATIVE" ......

(1) Standard ICL

Answer:
......
"CARRYOVER TO 1988 -- 0" is located on the right of "12, 500 (10- 15- 87)".
"RELEASED TO ACCTG" is located below of "AUTHORIZED COST:".
"PAID OUT OF 1987 BUDGET" is located on the right of "RELEASED TO ACCTG".
"CUMULATIVE" is located on the right of "PAID OUT OF 1987 BUDGET".
"JAN" is located below of "PAID OUT OF 1987 BUDGET" ....

(2) ICL-D3IE

(a)

Question: {text:"file",Box:[100 42 148 99]}{text:"PRODUCT",Box:[104 116 157 130]}
{text:"Triumph",Box:[199 116 249 127]}.....{text:"PROJECT TITLE",Box:[323 99 380 124]},
What are the labels for these texts?

MARKETING RESEARCH PROJECT APPROVAL
(To be filled out by Marketing Research Departmen
PRODUCT
Triumph
PROJECT TITLE
Triumph Disaster Check Study

Answer :
{text:"file",Box:[100 42 148 99],entity:other}
{text:"PRODUCT",Box:[104 116 157 130],entity:header} ✗
{text:"Triumph",Box:[199 116 249 127],entity:answer}
......
{text:"PROJECT TITLE",Box:[323 99 380 124],entity:header}
{text:"Triumph Disaster Check Study",Box:[399 111 594 126],entity:answer}

(1) Standard ICL Test

Answer:
{text:"file",Box:[100 42 148 99],entity:other}
{text:"PRODUCT",Box:[104 116 157 130],entity:question} ✓
{text:"Triumph",Box:[199 116 249 127],entity:answer}
......
{text:"PROJECT TITLE",Box:[323 99 380 124],entity:question}
{text:"Triumph Disaster Check Study",Box:[399 111 594 126],entity:answer}

(2) ICL-D3IE Test

(b)

Figure 6: Case study on comparison of (a) positional relationship description and (b) predictions generated by Standard ICL (Davinci-003) and ICL-D3IE (Davinci-003).

6. Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants (No. 62222203 and 61976049).

References

[1] 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. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2, 3

[2] Haoyu Cao, Xin Li, Jiefeng Ma, Deqiang Jiang, Antai Guo, Yiqing Hu, Hao Liu, Yinsong Liu, and Bo Ren. Query-driven generative network for document information extraction in the wild. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4261–4271, 2022. 1

[3] Rui CAO, Roy Ka-Wei LEE, Wen-Haw CHONG, and Jing JIANG. Prompting for multimodal hateful meme classification. 2022. 3

[4] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022. 2

[5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *ArXiv preprint*, abs/2204.02311, 2022. 2

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, pages 4171–4186, 2019. 5

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 6

[8] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 2

[9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. 2022. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[11] Zhangxuan Gu, Changhua Meng, Ke Wang, Jun Lan, Weiqiang Wang, Ming Gu, and Liqing Zhang. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *CVPR*, 2022. 2, 3, 5, 6

[12] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven CH Hoi. From images to textual prompts: Zero-shot vqa with frozen large language models. *arXiv preprint arXiv:2212.10846*, 2022. 2

[13] Teakgyu Hong, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *AAAI*, 2022. 1, 5, 6

[14] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*, 2022. 2

[15] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. *arXiv preprint arXiv:2204.08387*, 2022. 1, 2, 5, 6

[16] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In *ICDAR*, pages 1516–1520, 2019. 2, 5

[17] Guillaume Jaume, H. K. Ekenel, and J. Thiran. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. *ICDARW*, 2:1–6, 2019. 5

[18] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents. In *ICDARW*, 2019. 1, 2

[19] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022. 1

[20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL HLT*, 2016. 2

[21] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer. *arXiv preprint arXiv:2203.02378*, 2022. 1

[22] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *CVPR*, 2021. 1

[23] Yulin Li, Yuxi Qian, Yuechen Yu, Xiameng Qin, Chengquan Zhang, Yan Liu, Kun Yao, Junyu Han, Jingtuo Liu, and Errui Ding. Structext: Structured text understanding with multimodal transformers. In *ACM Multimedia*, 2021. 1

[24] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. *arXiv preprint arXiv:1903.11279*, 2019. 2

[25] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*, 2022. 2

[26] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021. 1

[27] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126, 2020. 6

[28] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing. In *Document Intelligence Workshop at Neural Information Processing Systems*, 2019. 1, 2

[29] Seunghyun Park, S. Shin, Bado Lee, Jihyo Lee, Jaeheung Surh, Minjoon Seo, and Hwal-Suk Lee. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019. 5

[30] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019. 4

[31] Omar Shaikh, Hongxin Zhang, William Held, Michael Bernstein, and Diyi Yang. On second thought, let's not think step by step! bias and toxicity in zero-shot reasoning. *arXiv preprint arXiv:2212.08061*, 2022. 2

[32] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *ArXiv preprint*, abs/2210.09261, 2022. 2

[33] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *arXiv preprint arXiv:2106.13884*, 2021. 3

[34] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*, 2022. 2

[35] Jiapeng Wang, Lianwen Jin, and Kai Ding. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. In *ACL*, 2022. 2, 5, 6

[36] Lei Wang, Jiabang He, Xing Xu, Ning Liu, and Hui Liu. Alignment-enriched tuning for patch-level pre-trained document image models. *arXiv preprint arXiv:2211.14777*, 2022. 2

[37] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 2

[38] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *CoRR*, abs/2206.07682, 2022. 2

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. In *Thirty-sixth Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022. 2

[40] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. In *LayoutLM: Pre-training of Text and Layout for Document Image Understanding*, pages 1192–1200, 2020. 2, 3, 5

[41] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *KDD*, 2020. 1, 6

[42] 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. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*, 2021. 1, 5, 6

[43] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089, 2022. 2, 3

[44] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022. 3

[45] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068, 2022. 2

[46] Xiaohui Zhao, Zhuo Wu, and Xiaoguang Wang. CUTIE: Learning to Understand Documents with Convolutional Universal Text Information Extractor. *arXiv preprint arXiv:1903.12363*, 2019. 2

[47] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022. 2