DocLLM

A LAYOUT-AWARE GENERATIVE LANGUAGE MODEL FOR MULTIMODAL DOCUMENT UNDERSTANDING

- Hrithik Sagar

Abstract

- Consideres spatial layout and textual semantics while avoiding image encoders
- Focus: BBOXES (for layout structure)
  - Cross alignment btw text and spatial modalities (by decomposing attention mechanism to set of disentangled matrices)
- Devices pretraining Obj to learn infill text segments
- Covers 4 document intelligence tasks
  - visual question answering (VQA),
  - natural language inference (NLI),
  - key information extraction (KIE), and
  - document classification (CLS)

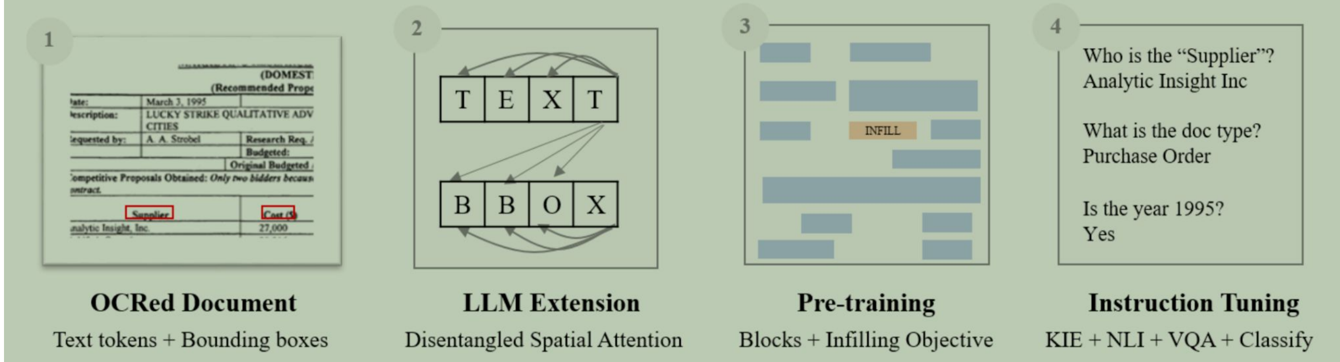

Figure 1: Key elements of `DocLLM`. (1) Input documents contain text tokens and their bounding boxes. (2) Attention mechanism of LLMs are extended to capture dependencies between text semantics and spatial layouts. (3) Infilling text blocks is used as pre-training objective. (4) Task adaptation is performed on a newly collated dataset of instructions.

Common characteristics of docs:

1. Heterogeneous content
2. Irregular layout
3. Disjoint texts segment

Modifications to pretraining objective to tackle issue:

- Issue: Preceding tokens may be irrelevant due to diverse text layouts—horizontal, vertical, or staggered.
- Modification 1: Using cohesive text blocks for context, and
- Modification 2: infilling by considering surrounding tokens.

Covers single and multiple pages doc

Framework:

1. A light-weight extension to LLMs designed for understanding visual documents.
2. A disentangled spatial attention mechanism that captures cross-alignment between text and layout modalities.
3. An infilling pre-training objective tailored to address irregular layouts effectively.
4. An instruction-tuning dataset specially curated towards visual document intelligence tasks.
5. Comprehensive experiments and valuable insights into the model behavior.

1. **Comparison of Model Types**:
    - **UDOP and LayoutLM**: These models **combine vision and language** (multimodal) and perform better than vision-only models.
    - **Donut and Pix2Struct**: These are **vision-only models**, meaning they rely purely on images to understand documents.
2. **Limitation of UDOP and LayoutLM**:
    - Even though they perform well, they require **task- and dataset-specific fine-tuning**.
    - This means they must be trained separately for different VRDU tasks, making them **less flexible**.
    - Due to this limitation, the authors **exclude** them from their analysis.
3. **Newer Models Built on LLMs**:
    - **mPLUG-DocOwl and UReader**: These models are based on **Large Language Models (LLMs)** and are trained using **instruction fine-tuning**.
    - They are trained on a **diverse mix** of:
        - VRDU datasets
        - Visual datasets
        - Textual datasets
    - Because of this, they show **strong zero-shot generalization**—meaning they can handle new tasks without additional fine-tuning.

docLLM Architecture (Common for LLMs)

LLM Architecture

```
Start
 |
 ▼
Input Processing
 |
 ├── Text Token Embedding (from input text)
 |
 ├── Spatial Token Embedding (from OCR bounding boxes)
 |
 ▼
Positional Encoding for Text & Spatial Features
 |
 ▼
Multi-Modal Transformer Block (Stacked)
 |
 ├── Multi-Head Self-Attention
 |      ├── Text-to-Text Attention
 |      ├── Spatial-to-Spatial Attention
 |      ├── Cross Attention (Text ↔ Spatial)
 |
 ├── Add & Normalize
 |
 ├── Feed-Forward Network
 |
 ├── Add & Normalize
 |
 ▼
Final Multi-Modal Transformer Output
 |
 ▼
Linear Layer & Softmax
 |
 ▼
Text Infilling Prediction (Self-Supervised Learning)
 |
 ▼
Generated Output (Partially Masked or Completed Text)
 |
 ▼
End
```

DocLLM Architecture

```
Start
 |
 ▼
Input Token Embedding
 |
 ▼
Positional Encoding
 |
 ▼
Transformer Block (Stacked)
 |
 ├── Multi-Head Self-Attention
 |
 ├── Add & Normalize
 |
 ├── Feed-Forward Network
 |
 ├── Add & Normalize
 |
 ▼
Final Transformer Block Output
 |
 ▼
Linear Layer & Softmax
 |
 ▼
Generated Token Output
 |
 ▼
Next Token Prediction (Auto-Regressive) ➔ Loop Back to Input
 |
 ▼
End
```

Why Modify the Pretraining Objective to Text Infilling?

1. **Better Contextual Understanding** – Uses both **prefix and suffix tokens** for more accurate predictions, unlike standard left-to-right models.

2. **Robustness to OCR Noise** – Helps correct missing or misaligned text in scanned documents.

3. **Improved Handling of Structured Data** – Learns relationships between fields in forms, invoices, and contracts.

4. **More Natural Completions** – Fills in logical gaps rather than just continuing text sequentially.

Infilling makes DocLLM **more accurate, resilient, and effective** for document understanding.

Template Design for Document Understanding (KIE & CLS)

- **Extraction Instructions** → Teach DocLLM to map **key names** in prompts to **document fields** for value retrieval.
- **Classification Instructions** → Help the model **understand key/document characteristics** for accurate classification.
- **MCQ Instructions** → Enable DocLLM to use key names or document types in prompts to:
  - Classify **extracted values** (for KIE).
  - Classify **entire documents** (for CLS).

Instruction-Tuning Data Mix for DocLLM

- **Visual Question Answering (VQA)**
  - Uses **DocVQA, WTQ, VisualMRC, DUDE, BizDocs2**.
  - One instruction template for **Supervised Fine-Tuning (SFT)**.
  - **Example:** *"{document} What is the deadline for scientific abstract submission for ACOG - 51st annual clinical meeting?"*
- **Natural Language Inference (NLI)**
  - Uses **TabFact** (only available DocAI NLI dataset).
  - **Example:** *"{document} 'The UN commission on Korea includes 2 Australians.' Yes or No?"*

- **Key Information Extraction (KIE)**
  - Uses **Kleister Charity (KLC), CORD, FUNSD, DeepForm, PWC, SROIE, VRDU ad-buy, BizDocs**.
  - Three instruction templates: **extraction, internal classification, MCQ**.
  - “None” answer added if the key doesn’t exist.
  - **Example:** *”{document} What is the value for the ‘charity number’?”*
- **Document Classification (CLS)**
  - Uses **RVL-CDIP, BizDocs**.
  - Two instruction templates: **internal classification, MCQ**.
  - **Downsamples RVL-CDIP** in training to balance datasets.
  - **Example:** *”{document} What type of document is this? Possible answers: [budget, form, file folder, questionnaire].”*

Models by DocLLM

1. DocLLM 1B
    a. Based on Falcon 1B architecture
    b. 24 layers, 16 attention heads and 1536 hidden size
    c.
2. DocLLM 7B
    a. Based on LLAMA 2 7B architecture
    b. 36 layers, 32 heads, and a hidden size of 4,096.
    c.

Table 4: Model configuration and training hyperparameters setting for `DocLLM-1B` and `-7B`.

	DocLLM-1B		**DocLLM-7B**	
Backbone	Falcon-1B [5]		Llama2-7B [4]	
Layers	24		36	
Attention heads	16		32	
Hidden size	1536		4096	
Precision	bfloat16		bfloat16	
Batch size	2		5	
Max context length	1,024		1,024	
	Pre-train	**Instruct-tune**	**Pre-train**	**Instruct-tune**
Learning rate	2×10^{-4}	1×10^{-4}	3×10^{-4}	1×10^{-4}
Warmups	1000	500	1000	500
Scheduler type	cosine	cosine	cosine	cosine
Weight decay	0.1	0.1	0.1	0.1
Adam βs	(0.9, 0.96)	(0.9,0.96)	(0.9,0.95)	(0.9,0.95)
Adam epsilon	1×10^{-5}	1×10^{-5}	1×10^{-6}	1×10^{-6}