A Appendix

A.1 Implementation Details

The hyperparameters we used for pre-training and fine-tuning downstream tasks are shown in Table 1. In general, our hyperparameter settings are similar to the one used in Deformable DETR [11]. Here, "base" configurations are those common for all experiments. It is worth mentioning that we use the box refinement design proposed in [11] which we find helpful for bounding box prediction.

For "pre-training", we use a three-stage pre-training. That is, in the first stage, we pre-train on 1M IIT-CDIP samples [6] for 20 epochs. Then, we pre-train on 5M samples for 5 epochs. In the final stage, we pre-train on 11M samples (full dataset) for 2 epochs.

"receipt parsing", "entity labeling" and "entity linking" show the settings we used to obtain the numbers we reported in Table 1 and 2 of the main manuscript. For "entity extraction", since we follow the existing work to address this task with IOB tagging (for fair comparison), we do not apply Eq. (2) and Eq. (3) (in main manuscript) as the loss function. But instead we simply use a cross-entry loss. Here "CE loss weight" is the weight of this loss.

config type	config name	value
	optimizer	AdamW [7]
	base LR	$2e^{-4}$
	cnn LR	$2e^{-5}$
base	language encoder LR.	$1e^{-5}$
	weight decay	$1e^{-4}$
	LR schedule	step
	box refinement [11]	yes
	batch size	32
pre-training	epochs	20, 5, 2
	training samples	1M, 5M, 11M
	LR drop step size	20
	batch size	8
	epochs	200
	LR drop step size	160
receipt parsing	EE loss weight	5.0
	EL loss weight	1.0
	anchor word	first+last
	primary anchor	name
	batch size	8
entity labeling	epochs	50
	LR drop step size	40
	CE loss weight	2.0
	batch size	8
	epochs	200
entity linking	LR drop step size	160
	EE loss weight	0.0
	EL loss weight	10.0

Table 1: Implementation settings. Here, "LR" stands for learning rate. "EE" stands for entity extraction. "EL" stands for entity linking. "CE" stands for cross-entropy.

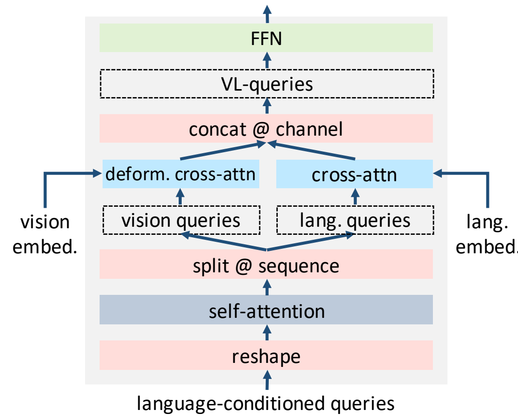


Figure 1: Architecture of a VL-decoder layer. It splits language-conditioned queries into two parts, vision queries and language queries. Vision queries extract visual information via deformable cross-attention [11]. Language queries extract linguistic information via language-conditioned cross-attention.

A.2 VL Decoder Layer

The VL-decoder has 6 layers, where each layer consists of a self-attention module, a deformable cross-attention module [11] and a standard cross-attention module [8] as shown Fig. 1. Let $\mathbf{Q} \in \mathbb{R}^{L \times D}$ be the input decoder queries. We first reshape $\mathbf{Q}$ to $\mathbf{Q}' \in \mathbb{R}^{2L \times \frac{D}{2}}$, where L is the number of input queries and D is the channel size. After the self-attention module, we split $\mathbf{Q}'$ into two equally sized queries $\mathbf{Q}^v \in \mathbb{R}^{L \times \frac{D}{2}}$ and $\mathbf{Q}^l \in \mathbb{R}^{L \times \frac{D}{2}}$. The vision queries $\mathbf{Q}^v$ extract visual features via deformable cross-attention. The language queries $\mathbf{Q}^l$ extract language features via cross-attention, where we apply Eq. (1) in the main text to assign explicit language semantics to queries. The outputs from the two cross-attention modules are concatenated at the channel dimension to recover the original shape, i.e., $\mathbf{Q}^{vl} \in \mathbb{R}^{L \times D} = \text{concat}(\mathbf{Q}^v, \mathbf{Q}^l)$. We use a fully connected layer at the end to further fuse vision and language information along the channel dimension.

A.3 Additional Results

Next, we show some additional experiments to further demonstrate our model's property under different settings. In the experiments, the default settings (that match Table 1) are marked in gray.

Comparison of using anchor words from different line-item fields as primary anchors. From Table 2, we can see that using the anchor word of "name" gives the best result. This is because in the test set, all the line-items contain this field. So, it is reliable to use this field as the primary anchor for entity linking. For the other fields, their performances

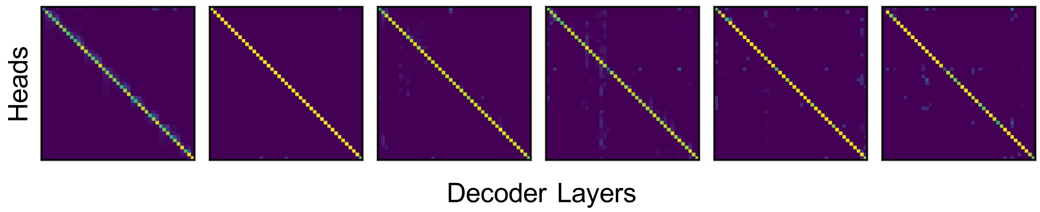


Figure 2: Visualization of language-conditioned cross-attention.

primary anchor	LI's with this field	parsing (CORD)
unit price	28%	66.5
count	90%	91.4
price	99%	94.0
name	100%	**94.4**
first	-	94.0

Table 2: Receipt parsing results using anchor words from different line-item fields as primary anchors. The middle column indicates the proportion of line-items that contain this field.

are lower when they do not frequently present in line-items. Also, note that the middle column only considers line-items not key-values. Therefore, it is possible that the parsing performance numbers are even higher than the proportion of line-items that contain this field.

Comparison of using predicted text and ground truth text as inputs. The existing works [4, 2, 9, 1] use ground truth text as the input in the experiments. We also follow the same way for fair comparison. However, it would be interesting to see how the models work if predicted text (from an OCR system) is used. In particular, using ground truth text as input is not in favor of vision only approaches such as Donut [5]. Table 3 shows the results of the comparison. In this experiment, we use an in-house OCR system which has comparable performances with the state-of-the-art OCR solutions (e.g., those from Azure, GCP or AWS). As we can see, there is a performance drop when we switch from using ground truth text to predicted text. However, compared with the vision only solution Donut, we are still noticeably better. This indicates the importance of having language inputs.

Model performance at different pre-training stages. In Table 4, each stage is based on the pre-trained model from its previous stage. For example, stage 2 initializes the model using the weights pre-trained from stage 1. As we can see, when more data is used, the model's performance continues improving on the FUNSD entity extraction task.

model	input text	parsing (CORD)
Donut [5]	none	87.8
SPADE [4]	gt	92.5
LayoutLMv2 [10]	pred.	92.2
BROS [2]	pred.	92.1
LayoutLMv3 [3]	pred.	93.0
DocTr (ours)	pred.	93.7
LayoutLMv2 [10]	gt	92.7
BROS [2]	gt	92.9
LayoutLMv3 [3]	gt	93.6
DocTr (ours)	gt	**94.4**

Table 3: Receipt parsing results comparison of using predicted text (from OCR system) and ground truth text as the input to the model. "gt" means ground truth text is used (but the text is ordered in raster scan manner). "pred." means predicted text is used. "none" means no text input is used.

stage	# samples	# epochs	F1
1	1M	20 epochs	82.1
2	5M	5 epochs	83.1
3	11M	2 epochs	**84.0**

Table 4: Entity extraction comparison results at different pre-training stages.

# epochs	F1
100	93.8
200	94.4
300	**94.5**

Table 5: Receipts parsing results using different number of epochs.

Model performance at different number of epochs. In Table 5, we show model's performance when fine-tuned with different number of epochs. We notice that the model's performance tends to plateau after 200 epochs.

Visualization of language-conditioned cross-attention. We further verify the behavior of this cross-attention mechanism by visualizing the cross-attention matrices. The cross-attention results are extracted from the cross-attention module of each decoder layer, i.e., we check the cross-attention between the language inputs and the language-conditioned queries. As we can see in Fig 2, the attention weights are high on the diagonal of the attention matrices. This shows that we successfully established this one-to-one mapping between the queries (and thus decoder outputs) and language tokens.

Additional visualizations. Fig 3-6 show additional comparisons of three structured information extraction formulations. Fig 7 show two receipts parsing failure cases. Fig 8-11 show additional pre-training outputs using our proposed masked detection modeling task.

References

[1] Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*, 2021. 2

[2] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10767–10775, 2022. 2

[3] 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. 2

[4] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. Spatial dependency parsing for semi-structured document information extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 330–343, 2021. 2

[5] 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*, pages 498–517. Springer, 2022. 2

[6] David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666, 2006. 1

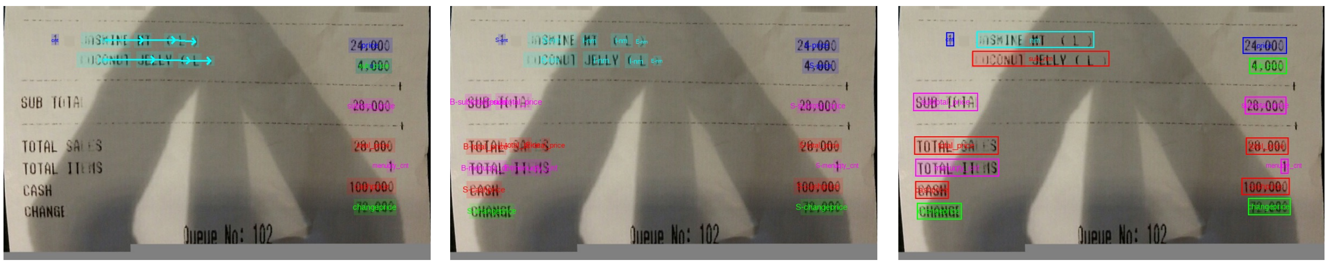
[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 1

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[9] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020. 2

[10] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, 2021. 2

[11] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 1



[
  [
    {"menu.nm": "JASMINE MT (L)"},
    {"menu.cnt": "1"},
    {"menu.price": "24,000"},
    ~~{"menu.sub_nm": "COCONUT JELLY (L)"},~~
    ~~{"menu.sub_price": "4,000"}~~
  ],
  [
    {"menu.nm": "COCONUT JELLY (L)"},
    {"menu.price": "4,000"}
  ],
  [
    {"sub_total.subtotal_price": "28,000"}
  ],
  [
    {"total.total_price": "28,000"},
    {"total.menuqty_cnt": "1"},
    {"total.cashprice": "100,000"},
    {"total.changeprice": "72,000"}
  ]
]

(a) IOB Tagging

[
  [
    {"menu.nm": "JASMINE MT (L)"},
    {"menu.cnt": "1"},
    {"menu.price": "24,000"},
    ~~{"menu.sub_nm": "COCONUT JELLY (L)"},~~
    ~~{"menu.sub_price": "4,000"}~~
  ],
  [
    {"menu.nm": "COCONUT JELLY (L)"},
    {"menu.price": "4,000"}
  ],
  [
    {"sub_total.subtotal_price": "28,000"}
  ],
  [
    {"total.total_price": "28,000"},
    {"total.menuqty_cnt": "1"},
    {"total.cashprice": "100,000"},
    {"total.changeprice": "72,000"}
  ]
]

(b) SPADE

[
  [
    {"menu.cnt": "1"},
    {"menu.nm": "JASMINE MT (L)"},
    {"menu.price": "24,000"},
    {"menu.sub_nm": "COCONUT JELLY (L)"},
    {"menu.sub_price": "4,000"}
  ],
  [
    {"sub_total.subtotal_price": "28,000"}
  ],
  [
    {"total.cashprice": "100,000"},
    {"total.changeprice": "72,000"},
    {"total.menuqty_cnt": "1"},
    {"total.total_price": "28,000"}
  ]
]

(c) Ours

Figure 3: CORD receipt parsing results on **receipt_00002** sample. Each result consists of the visualization of model predictions, and the parsing outputs. **(a)** IOB tagging visualizes the predicted tags of OCR words. **(b)** SPADE visualizes the decoded graph, and arrows between words indicate that the words are linked in the same entity. **(c)** Ours visualizes the predicted anchor words and their bounding boxes. For simplicity of the visualization, entity-linking results are not visualized in here. For the parsing outputs, green/red text means the predicted text matches/does not match ground truth. ~~Strikethrough~~ text means the ground truth text is missed from prediction. Best view in color and zoom-in for details of the visualization. For this example, both IOB tagging and SPADE recognized the second row of the line-item as an individual line-item. DocTr understands the line-item better by recognizing it as a single line-item.

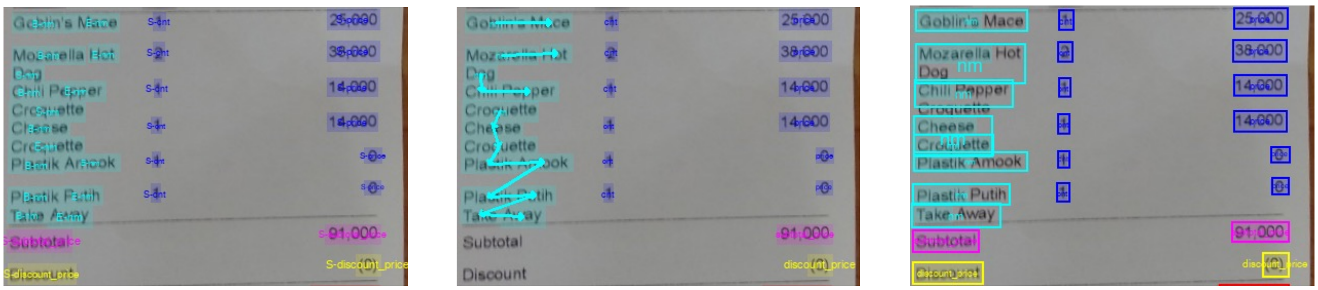
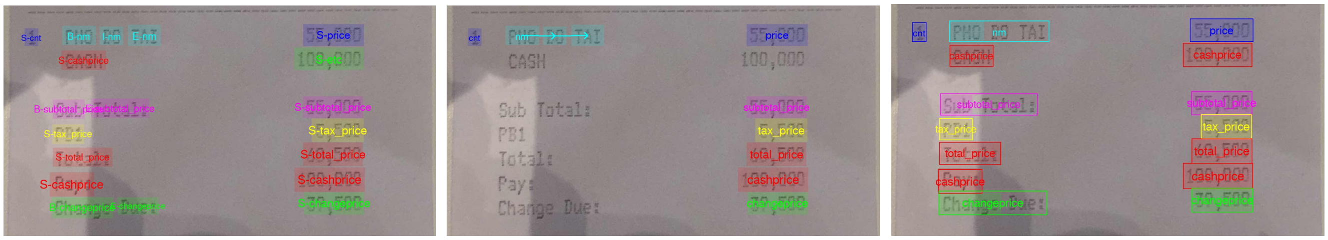


Figure 4: CORD receipt parsing results on **receipt_00017** sample. Each result consists of the visualization of model predictions, and the parsing outputs. **(a)** IOB tagging visualizes the predicted tags of OCR words. **(b)** SPADE visualizes the decoded graph, and arrows between words indicate that the words are linked in the same entity. **(c)** Ours visualizes the predicted anchor words and their bounding boxes. For simplicity of the visualization, entity-linking results are not visualized in here. For the parsing outputs, green/red text means the predicted text matches/does not match ground truth. ~~Strikethrough~~ text means the ground truth text is missed from prediction. Best view in color and zoom-in for details of the visualization. This is a challenging sample with several line-items. Some line-items have single-line name, and some have two-line name. As we can see, SPADE totally failed in this case, with a wrong graph for the line-item names. IOB tagging and DocTr are better and detected names mostly correct.



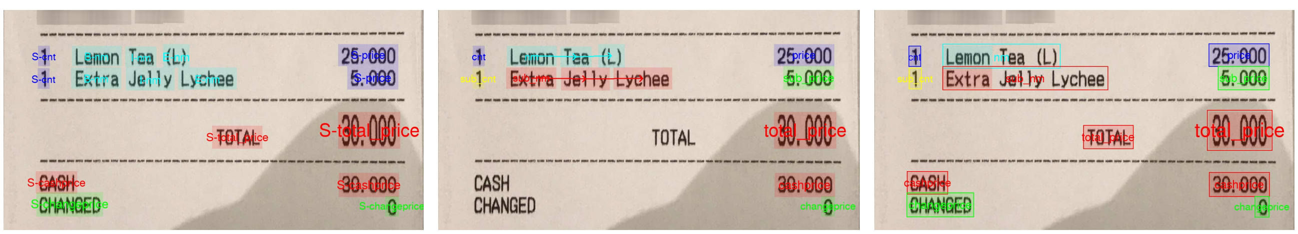
(a) IOB Tagging
[
[
{"menu.cnt":"1"},
{"menu.nm":"PHO BO TAI"},
{"menu.price":"55,000"}
],
[
{"sub_total.subtotal_price":"55,000"},
{"sub_total.tax_price":"5,500"}
],
[
~~{"total.cashprice":"100,000"}~~,
{"total.cashprice":"100,000"},
{"total.changeprice":"39,500"},
{"total.total_price":"60,500"}
]
]

(b) SPADE
[
[
{"menu.cnt":"1"},
{"menu.nm":"PHO BO TAI"},
{"menu.price":"55,000"}
],
[
{"sub_total.subtotal_price":"55,000"},
{"sub_total.tax_price":"5,500"}
],
[
~~{"total.cashprice":"100,000"}~~,
{"total.cashprice":"100,000"},
{"total.changeprice":"39,500"},
{"total.total_price":"60,500"}
]
]

(c) Ours
[
[
{"menu.cnt":"1"},
{"menu.nm":"PHO BO TAI"},
{"menu.price":"55,000"}
],
[
{"sub_total.subtotal_price":"55,000"},
{"sub_total.tax_price":"5,500"}
],
[
{"total.cashprice":"100,000"},
{"total.cashprice":"100,000"},
{"total.changeprice":"39,500"},
{"total.total_price":"60,500"}
]
]

(a) IOB Tagging (b) SPADE (c) Ours

Figure 5: CORD receipt parsing results on **receipt_00058** sample. Each result consists of the visualization of model predictions, and the parsing outputs. **(a)** IOB tagging visualizes the predicted tags of OCR words. **(b)** SPADE visualizes the decoded graph, and arrows between words indicate that the words are linked in the same entity. **(c)** Ours visualizes the predicted anchor words and their bounding boxes. For simplicity of the visualization, entity-linking results are not visualized in here. For the parsing outputs, green/red text means the predicted text matches/does not match ground truth. ~~Strikethrough~~ text means the ground truth text is missed from prediction. Best view in color and zoom-in for details of the visualization. In this example, "CASH" is a subtotal. But it is right below the line-item. Thus, both "IOB Tagging" and "SPADE" missed this detection.



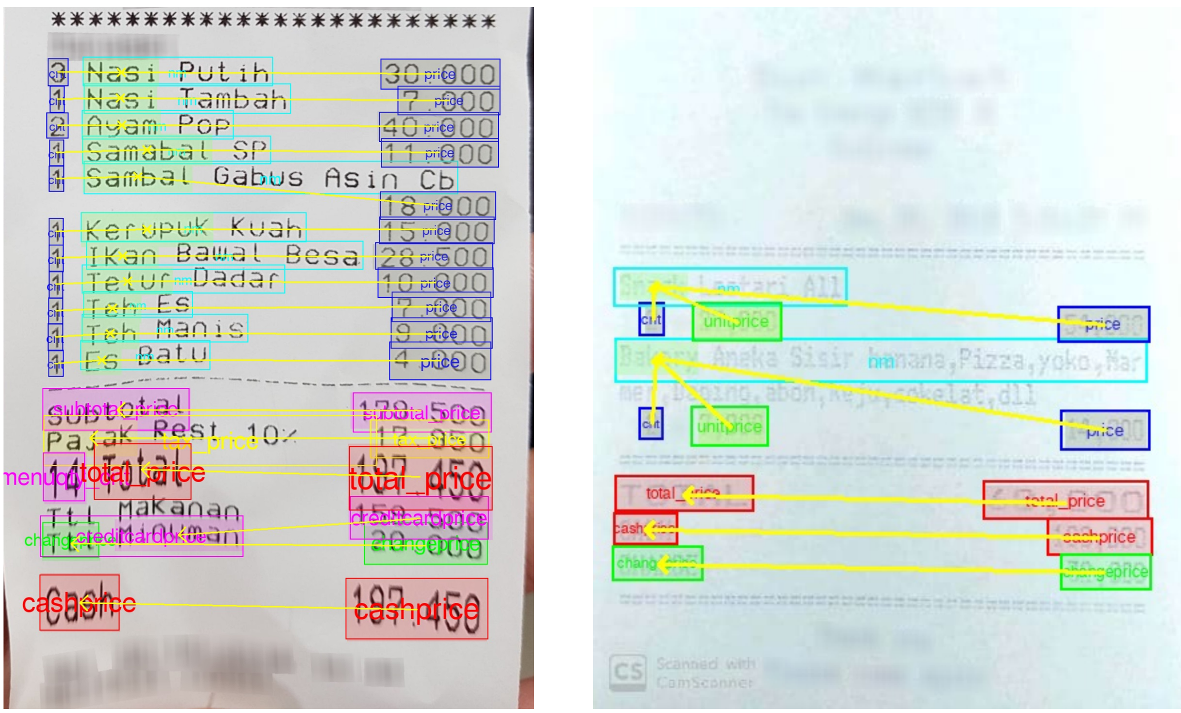
(a) IOB Tagging
[
[
{ "menu.cnt": "1" },
{ "menu.nm": "Lemon Tea (L)" },
{ "menu.price": "25.000" },
~~{ "menu.sub_cnt": "1" }~~,
~~{ "menu.sub_nm": "Extra Jelly Lychee" }~~,
~~{ "menu.sub_price": "5.000" }~~
],
[
{ "menu.cnt": "1" },
{ "menu.nm": "Extra Jelly Lychee" },
{ "menu.price": "5.000" }
],
[
{ "total.cashprice": "30.000" },
{ "total.changeprice": "0" },
{ "total.total_price": "30.000" }
]
]

(b) SPADE
[
[
{ "menu.cnt": "1" },
{ "menu.nm": "Lemon Tea (L)" },
{ "menu.price": "25.000" },
{ "menu.sub_cnt": "1" },
{ "menu.sub_nm": "Extra Jelly Lychee" },
{ "menu.sub_price": "5.000" }
],
[
{ "total.cashprice": "30.000" },
{ "total.changeprice": "0" },
{ "total.total_price": "30.000" }
]
]

(c) Ours
[
[
{ "menu.cnt": "1" },
{ "menu.nm": "Lemon Tea (L)" },
{ "menu.price": "25.000" },
{ "menu.sub_cnt": "1" },
{ "menu.sub_nm": "Extra Jelly Lychee" },
{ "menu.sub_price": "5.000" }
],
[
{ "total.cashprice": "30.000" },
{ "total.changeprice": "0" },
{ "total.total_price": "30.000" }
]
]

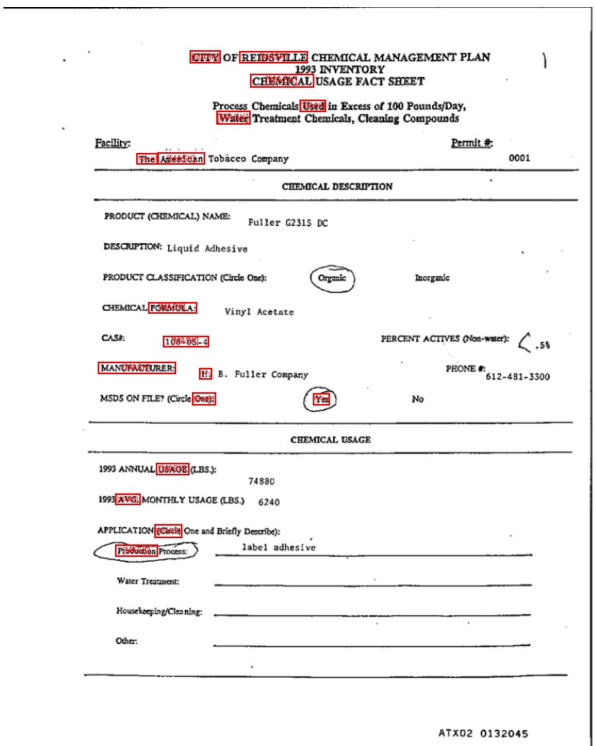
(a) IOB Tagging (b) SPADE (c) Ours

Figure 6: CORD receipt parsing results on **receipt_00078** sample. Each result consists of the visualization of model predictions, and the parsing outputs. **(a)** IOB tagging visualizes the predicted tags of OCR words. **(b)** SPADE visualizes the decoded graph, and arrows between words indicate that the words are linked in the same entity. **(c)** Ours visualizes the predicted anchor words and their bounding boxes. For simplicity of the visualization, entity-linking results are not visualized in here. For the parsing outputs, green/red text means the predicted text matches/does not match ground truth. ~~Strikethrough~~ text means the ground truth text is missed from prediction. Best view in color and zoom-in for details of the visualization.

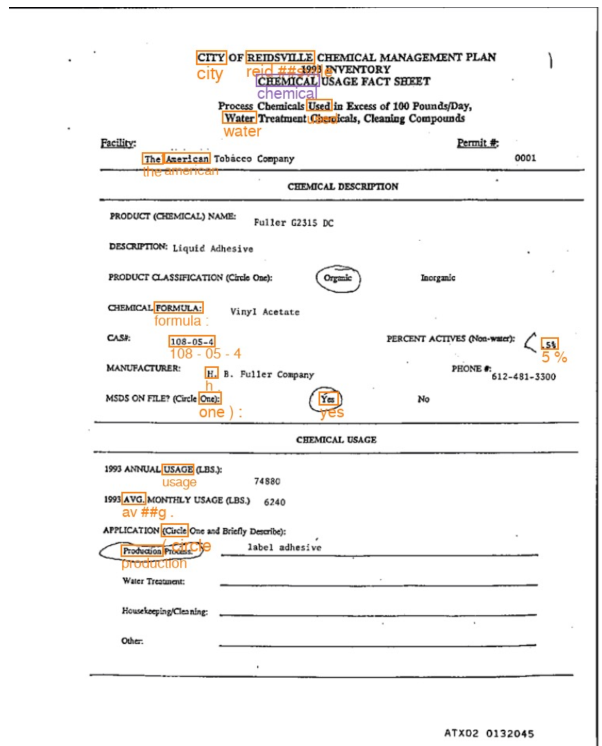


(a) receipt_00085 (b) receipt_00040

Figure 7: CORD receipt parsing failure cases. We visualize the predicted anchor words, their bounding boxes, and anchor word associations (yellow arrows).

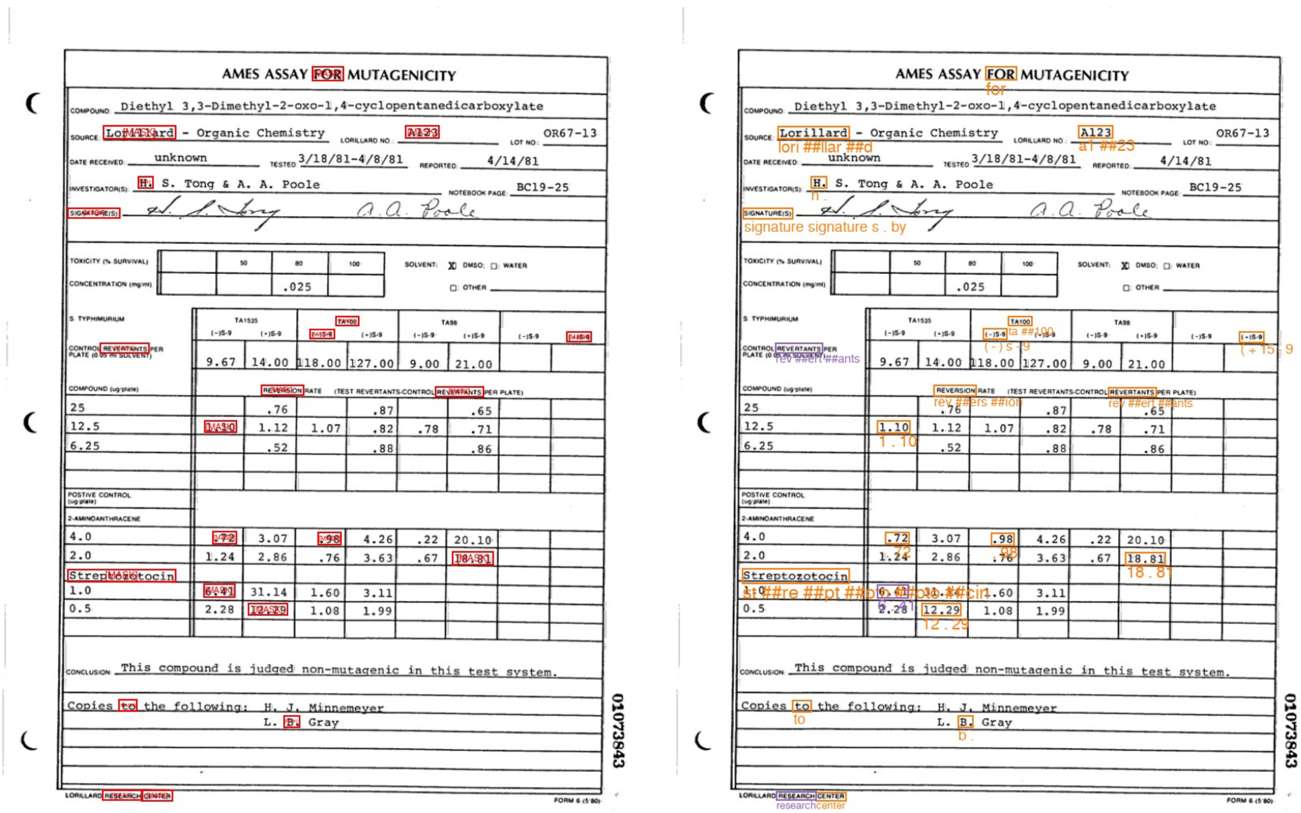


(a) masked input



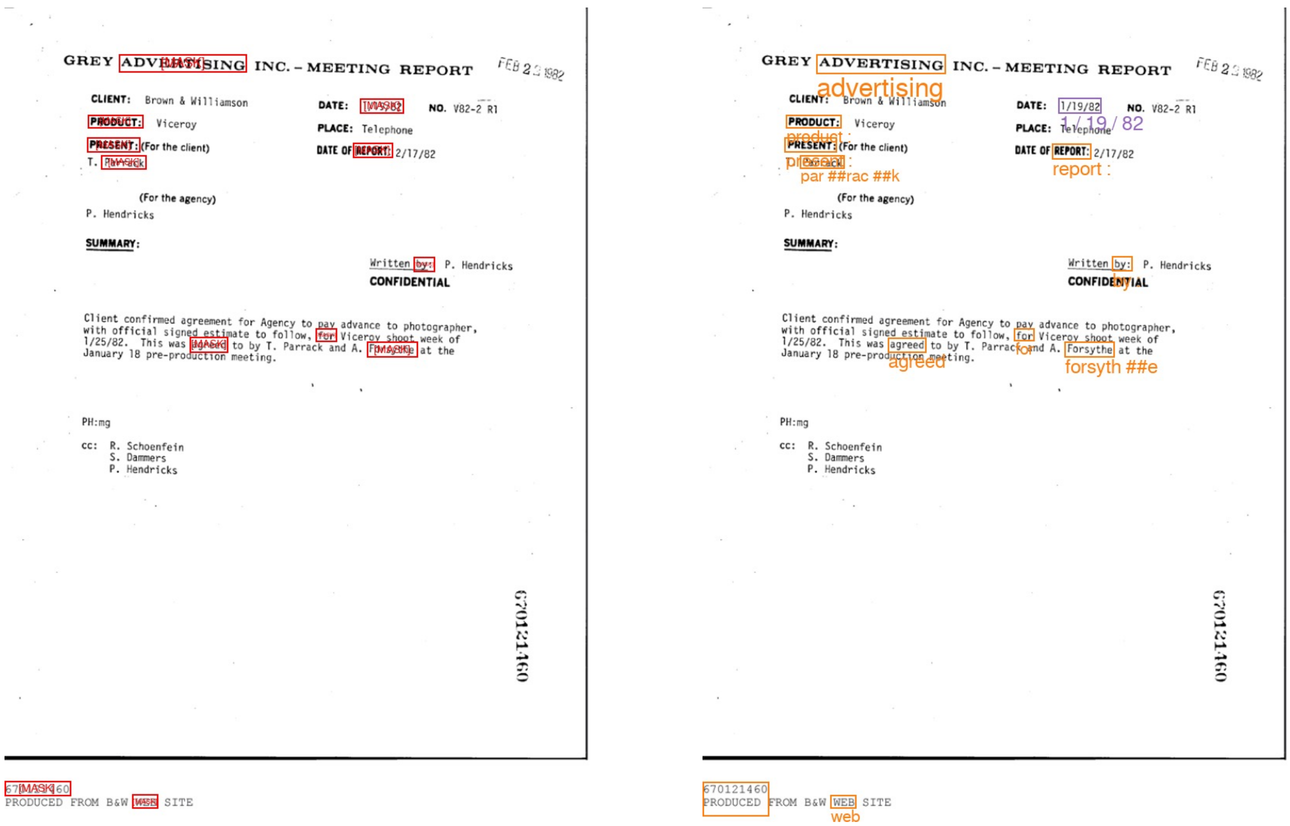
(b) predicted boxes and tokens

Figure 8: Example pre-training predictions on FUNSD sample **0060024314**. For inputs, we visualize masked word boxes, and their text is replace by `[MASK]`. For predictions, we visualize the predicted word boxes of the masked inputs. Under each box prediction, we also visualize its corresponding word token predictions.



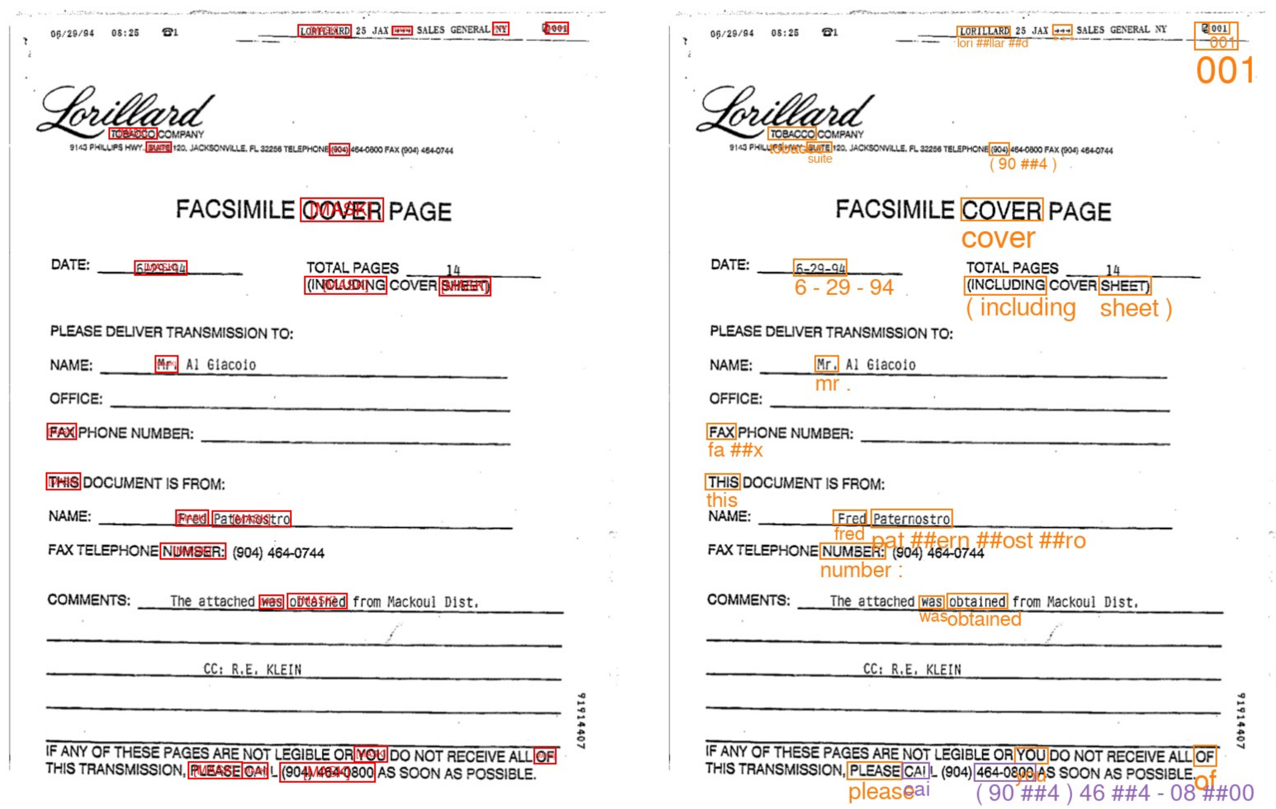
(a) masked input (b) predicted boxes and tokens

Figure 9: Example pre-training predictions on FUNSD sample **01073843**. For inputs, we visualize masked word boxes, and their text is replace by `[MASK]`. For predictions, we visualize the predicted word boxes of the masked inputs. Under each box prediction, we also visualize its corresponding word token predictions.



(a) masked input (b) predicted boxes and tokens

Figure 10: Example pre-training predictions on FUNSD sample **660978**. For inputs, we visualize masked word boxes, and their text is replace by `[MASK]`. For predictions, we visualize the predicted word boxes of the masked inputs. Under each box prediction, we also visualize its corresponding word token predictions.



(a) masked input (b) predicted boxes and tokens

Figure 11: Example pre-training predictions on FUNSD sample **91914407**. For inputs, we visualize masked word boxes, and their text is replace by `[MASK]`. For predictions, we visualize the predicted word boxes of the masked inputs. Under each box prediction, we also visualize its corresponding word token predictions.