

Supplementary Materials: PEneo: Unifying Line Extraction, Line Grouping, and Entity Linking for End-to-end Document Pair Extraction

Anonymous Authors

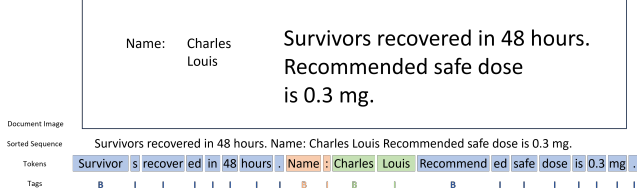


Figure 1: Example of BIO tagging in the SER+RE pipeline.

1 DETAILS OF THE SER+RE BASELINE

When training the SER model, we sorted the input lines with Augmented XY Cut [1] at the pre-processing step. If adjacent lines within an entity are sorted into neighboring positions, we organize their BIO tags at entity level. Labels of other lines were kept at line level. As shown in Figure 1, the two text lines of entity *Charles Louis* are sorted to adjacent positions, hence we tag them at the entity level. For entity *Survivors recovered in ...*, the last two lines are correctly arranged, while the first line is split out. In this case, we tag the last two lines as an entity, while the first line is tagged at line level. This setting helps detect all the content for those multi-line entities to the greatest extent.

For the RE module, we directly take the entity-level OCR results as input during the training phase, which is consistent with the settings of previous studies. During the inference phase, the RE model takes the output of the aforementioned SER step for linking prediction.

Performances of each sub-task in the SER+RE pipelines are shown in Table 1. Results of SER are evaluated based on the Augmented XY Cut sorted BIO tags, hence the values only roughly reflect the model’s SER capability and cannot be regarded as an accurate evaluation metric. The results also demonstrate that the SER+RE pipeline suffers from various instabilities. For example, on RFUND-EN, LiLT[InfoXLM]_{BASE} has a better performance on both sub-tasks than LiLT[EN-R]_{BASE}, but a lower score on pair extraction. We speculate that this may be caused by the differences in SER errors and the variation of the RE model’s sensitivity. Overall, the properties of the SER+RE pipeline remain to be explored.

2 INFLUENCE OF MODELING GRANULARITY

LiLT [5] and LayoutLMv3 [3] utilize entity-level boxes for layout embedding, while conventional settings ([4, 6, 7]) use word-level information. In our experiments, all the models are expected to take line-level boxes as input, which may affect their performance to some extent. To further illustrate the impact of different modeling granularity, we evaluate the SER performance of these models on FUNSD [2], using different types of bounding boxes. Results in Table 2 show that both LiLT and LayoutLMv3 suffer from performance

Table 1: Performance of each sub-task in the SER+RE pipeline. LiLT-I refers to LiLT[InfoXLM]_{BASE}, LiLT-R refers to LiLT[EN-R]_{BASE}, LaLM2B refers to LayoutLMv2_{BASE}, LaXLMB refers to LayoutXLM_{BASE}, LaLM3B refers to LayoutLMv3_{BASE}, and GeLaLM refers to GeoLayoutLM.

Dataset	Model	SER F1	RE F1	Pair F1
RFUND-EN	LiLT-I	80.28	67.18	52.18
	LiLT-E	79.66	65.25	54.33
	LaLM2B	84.57	61.30	49.06
	LaXLMB	80.83	66.95	52.98
	LaLM3B	86.05	69.22	57.66
	GeLaLM	92.90	87.73	69.03
RFUND-ZH	LiLT-I	91.78	77.51	66.50
	LaXLMB	92.54	73.50	64.11
	LaLM3B	90.20	81.63	72.14
RFUND-JA	LiLT-I	79.62	66.95	43.98
	LaXLMB	80.18	58.65	40.21
RFUND-ES	LiLT-I	84.98	77.12	63.85
	LaXLMB	86.72	81.01	66.75
RFUND-FR	LiLT-I	83.43	71.57	62.60
	LaXLMB	85.50	76.74	67.98
RFUND-IT	LiLT-I	82.59	68.53	60.57
	LaXLMB	85.05	65.17	63.04
RFUND-DE	LiLT-I	82.27	70.61	55.13
	LaXLMB	82.79	74.77	58.77
RFUND-PT	LiLT-I	83.23	67.27	52.96
	LaXLMB	85.09	60.86	59.79
SIBR	LiLT-I	92.90	89.00	72.76
	LaXLMB	93.61	81.99	70.45
	LaLM3B	93.50	87.07	73.51

Table 2: Influence of different modeling granularity. * are the results from the model’s original paper.

Model	Box Level	SER F1
LiLT[InfoXLM] _{BASE}	entity	84.15*
	word	73.78
LayoutXLM _{BASE}	entity	88.08
	word	79.40*
LayoutLMv3 _{BASE}	entity	90.29*
	word	79.96

drop when using the word-level information, indicating that the two models may underperform with fine-grained coordinates.

Algorithm 1 Pseudo code of the linking parsing algorithm**Input:** Prediction matrices $M^{(le)}$, $M^{(elh)}$, $M^{(elt)}$, $M^{(lgh)}$, $M^{(lgt)}$; Score matrices $P^{(le)}$, $P^{(elh)}$, $P^{(elt)}$, $P^{(lgh)}$, $P^{(lgt)}$.**Output:** List of parsed key-value pairs V .

```

1: Initialize dict  $D^{(le)}$ ,  $D^{(lgh)}$ ,  $D^{(lgt)}$ ,  $D^{(elh)}$ ,  $D^{(elt)}$ .
2: Initialize list  $V$  for storing parsed key-value pairs
3: for  $*$  in  $[(le), (lgh), (lgt), (elh), (elt)]$  do
4:   for  $i, j$  in all possible indices do
5:     if  $M^{(*)}[i][j] = 1$  and  $P^{(*)}[i][j][1] > D^{(*)}[i][1]$  then
6:        $D^{(*)}[i] = (j, P^{(*)}[i][j][1])$  ▷ save indices and scores
7:     end if
8:   end for
9:   for  $k, v$  in  $D^{(*)}$ .items() do
10:     $D^{(*)}[k] = v[0]$  ▷ remove the scores for clarity
11:   end for
12: end for
13: for  $t_{keh}, t_{veh}$  in  $D^{(elh)}$ .items() do ▷  $t_{keh}$ : head token of key's first-line.  $t_{veh}$ : head token of value's first-line
14:    $t_{clh} = t_{keh}$  ▷  $t_{clh}$ : head token of the current line
15:   Initialize list  $L_{key}$  to store all tokens of the key entity
16:   Initialize list  $L_{value}$  to store all tokens of the value entity
17:   if  $t_{clh}$  in  $D^{(le)}$ .keys() then ▷ get the tail token of current line  $t_{clt}$  from  $D^{(le)}$ 
18:      $t_{clt} = D^{(le)}[t_{clh}]$ 
19:   else
20:     continue ▷ discard invalid prediction
21:   end if
22:    $L_{key}$ .append(tokens in  $(t_{clh}, t_{clt})$ )
23:   while  $t_{clh}$  in  $D^{(lgh)}$ .keys() do ▷ get all tokens of the key entity
24:      $t_{nlh} = D^{(lgh)}[t_{clh}]$  ▷  $t_{nlh}$ : head token of next line
25:     if  $t_{clt}$  in  $D^{(lgt)}$ .keys() then
26:        $t_{nlt} = D^{(lgt)}[t_{clt}]$  ▷  $t_{nlt}$ : tail token of next line
27:     else
28:       break
29:     end if
30:     if  $(t_{nlh}, t_{nlt})$  in  $D^{(le)}$  then ▷ check line validity
31:        $L_{key}$ .append(tokens in  $(t_{nlh}, t_{nlt})$ )
32:     else
33:       break
34:     end if
35:      $t_{clh} = t_{nlh}$ 
36:   end while
37:   Repeat the above steps for  $t_{veh}$  and obtain  $L_{value}$ 
38:    $t_{ket} = L_{key}[-1]$  ▷  $t_{keh}$ : tail token of key's last-line.
39:    $t_{vet} = L_{value}[-1]$  ▷  $t_{veh}$ : tail token of value's last-line
40:   if  $D^{(elt)}[t_{ket}] == t_{vet}$  then ▷ check validity using the last tokens of the key and value entity
41:      $V$ .append( $(L_{key}, L_{value})$ )
42:   end if
43: end for
44: return  $V$ 

```

3 LINKING PARSING ALGORITHM

The algorithm flow of the linking parsing module is shown in Algorithm 1.

REFERENCES

- [1] Zhangxuan Gu, Changhua Meng, Ke Wang, Jun Lan, Weiqiang Wang, Ming Gu, and Liqing Zhang. 2022. XYLayoutLM: Towards Layout-aware Multimodal Networks for Visually-rich Document Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4583–4592.
- [2] Jean-Philippe Thiran Guillaume Jaume, Hazim Kemal Ekenel. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *ICDAR-OST*.
- [3] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proceedings of the 30th ACM International Conference on Multimedia*. 4083–4091.
- [4] Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. 2023. GeoLayoutLM: Geometric Pre-training for Visual Information Extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7092–7101.
- [5] Jiapeng Wang, Lianwen Jin, and Kai Ding. 2022. LiLT: A Simple yet Effective Language-Independent Layout Transformer for Structured Document Understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7747–7757.
- [6] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. LayoutXLM: Multimodal Pre-training for Multilingual Visually-rich Document Understanding. arXiv:2104.08836 [cs.CL]
- [7] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2021. 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)*. 2579–2591.