Towards Few-shot Entity Recognition in Document Images: A Label-aware Sequence-to-Sequence Framework

Anonymous ACL submission

Abstract

Entity recognition is a fundamental task in understanding document images. Traditional sequence labeling framework requires extensive datasets and high-quality annotations, which are typically expensive in practice. In this paper, we aim to build an entity recognition model based on only a few shots of annotated document images. To overcome the data limitation, we propose to leverage the label surface names to better inform the model of the target entity semantics. Specifically, we go beyond sequence labeling and develop a novel label-aware seq2seq framework, LASER. We design a new labeling scheme that generates the label surface names word-by-word explicitly after generating the entities. Moreover, we design special layout identifiers to capture the spatial correspondence between regions and labels. During training, LASER refines the label semantics by updating the label surface name representations and also strengthens the label-region correlation. In this way, LASER recognizes the entities from document images through both semantic and layout correspondence. Extensive experiments on two benchmark datasets demonstrate the superiority of LASER under the few-shot setting.

1 Introduction

Entity recognition lies in the foundation of document image understandings, which aims at extracting word spans that perform certain roles from the document images, such as *header*, *question*. Distinct from the text-only named entity recognition task, the document images, such as forms, tables, receipts, and multi-columns, provide a perfect scenario to apply multi-modal techniques into practice where the rich layout formats in such document images serve as the new, complementary signals for entity recognition performance in addition to the existing textual data.

Recent methods (Xu et al., 2020; Hong et al., 2020; Garncarek et al., 2021) follow the traditional sequence labeling framework to extract the word spans using the standard IOBES tagging schemes (Marquez et al., 2005; Ratinov and Roth, 2009) in named entity recognition tasks. These methods largely extend the label space by including combinations of the boundary identifiers (B, I, E, S) and entity types. For instance, when there are 3 target entity types, the extended label space would have 13 (i.e., $4 \times 3 + 1$) dimensions. As a result, they require extensive datasets and high-quality annotations to inform the model of the label semantics. Document images typically include various formats and have a high density of entities within each page. Given the complexity of the document images, it is expensive or almost impossible to acquire enough annotated data in certain application scenarios. Moreover, when it comes to the receipts or consent forms, ethical concerns would arise, making it hard to collect enough data.

To overcome the data limitation and better save human labors, we resort to build entity recognition models for document images under the few-shot setting, which means the models can only learn from a limited number of training pages and are generalized on new pages. In our method, we go beyond the sequence labeling framework and reformulate the entity recognition as a sequence-to-sequence task. Specifically, we propose a new generative labeling scheme for entity recognition — the label surface name is generated right after each entity as a part of the target sequence. In this way, different entity types are no longer independent dimensions in the label space. Instead, one can benefit from pre-trained language models to explicitly understand the semantic meanings of entity types from the label surfaces.

To this end, we propose a label-aware sequence-to-sequence framework for entity recognition, LASER. As shown in Figure 1, it follows our proposed generative labeling scheme to better solve the few-shot entity recognition task for document images. We implement LASER based on pre-trained language model LayoutReader (Wang et al., 2021), which is a layout-aware pre-trained sequence-to-sequence model. The semantic correlation is latent within the pre-trained models and we explicitly use it in sequence-to-sequence training to infer the correspondence between the entities and the label surface names. Specifically, after generating certain word spans, the model can choose to generate either the following words in the source sequence or label surface names. Generation probability of labels conditioned on entities can be learned through maximizing the log likelihood.

LASER also explicitly learns the spatial correspondence between label surface names and the entities. Given the document images, inputs from both text and layout formats are available. Compared with textual data, layout formats are easy to learn and can provide stronger signals in relation extraction and reading order detection (Wang et al., 2021, 2020). We design special embeddings as layout identifiers for the label surface names, so the generation probability of the next token is also aware of the correlation between the layout identifiers and the spatial embeddings of the entities. We intend to learn which areas the entities of certain labels are more likely to appear.

Considering both semantic and spatial correspondence between labels and entities, LASER is able to effectively recognize entities in document images with only a limited number of training samples. In contrast, the existing sequence labeling models fail to leverage the semantics or layout formats in an explicit way, thus requiring more data to learn the correlation between labels and entities.

We validate LASER using two benchmarks, FUNSD (Guillaume Jaume, 2019) and CORD-Lv1 (Park et al., 2019). Both datasets are from real scenarios and fully-annotated with textual contents and bounding boxes. We compare our model with strong baselines and study the label-entity semantic and spatial correlations. We summarize our contribution as follows.

- We reformulate the entity recognition task and propose a new generative labeling scheme that embeds the label surface names into the target sequence to explicitly inform the model of the label semantics.
- We propose a novel label-aware sequence-to-sequence framework LASER to better handle few-shot entity recognition tasks for document images than the traditional sequence labeling framework using both label semantics and layout format learning.
- Extensive experiments on two benchmark datasets demonstrate the effectiveness of LASER under few-shot settings.

Reproducibility. We will release the code and datasets on Github[1].

2 Problem Formulation

The few-shot entity recognition in the document images is to take the text and layout inputs from a limited number of training samples to predict the boundary of each entity and classify the entity into categories. Given a document image page $\mathcal{P}$, the words within the page are annotated with their textual contents w and the bounding boxes $B = (x_0, y_0, x_1, y_1)$ (top-left and bottom-right corners) serving as the inputs from textual and layout modalities. All the words and bounding boxes correspond to each other and are listed in a sequence so the entities are spans of these words referring to precise concepts. We randomly select a small set of training samples and evaluate the performance under the k-shot training, where k denotes the number of the training sample.

3 Our Generative Labeling Scheme

We propose our labeling scheme of entity recognition in the generative manner which generates the entity boundaries and the label surface names explicitly. Specifically, given an entity $e = [w_i, w_{i+1}..., w_j]$, we use the `[B]` and `[E]` to denote the boundary of the entity and append the label surface name afterwards. Overall, the generative formulation is to generate:

$$w_{i-1}, \texttt{\textbf{[B]}}, w_i, ..., w_j, \texttt{\textbf{[E]}}, \tau_{\mathbf{1}}, ..., \tau_{\mathbf{k}}, \texttt{\textbf{[T]}}, w_{j+1}$$

where `[B]` and `[E]` denote the start and end of the entity; $\tau_1...\tau_k$ are the words in the label surface name; `[T]` denotes the end of the label surface name. For example, "*Sender*" and "*Charles Duggan*" are a pair of *question* and *answer* from a document image. According to the generative labeling scheme, the corresponding generated sequence is that: `[B]` *Sender* `[E]` *question* `[T]` `[B]` *Charles Duggan* `[E]` *answer* `[T]`.

[1] https://github.com/anonymous

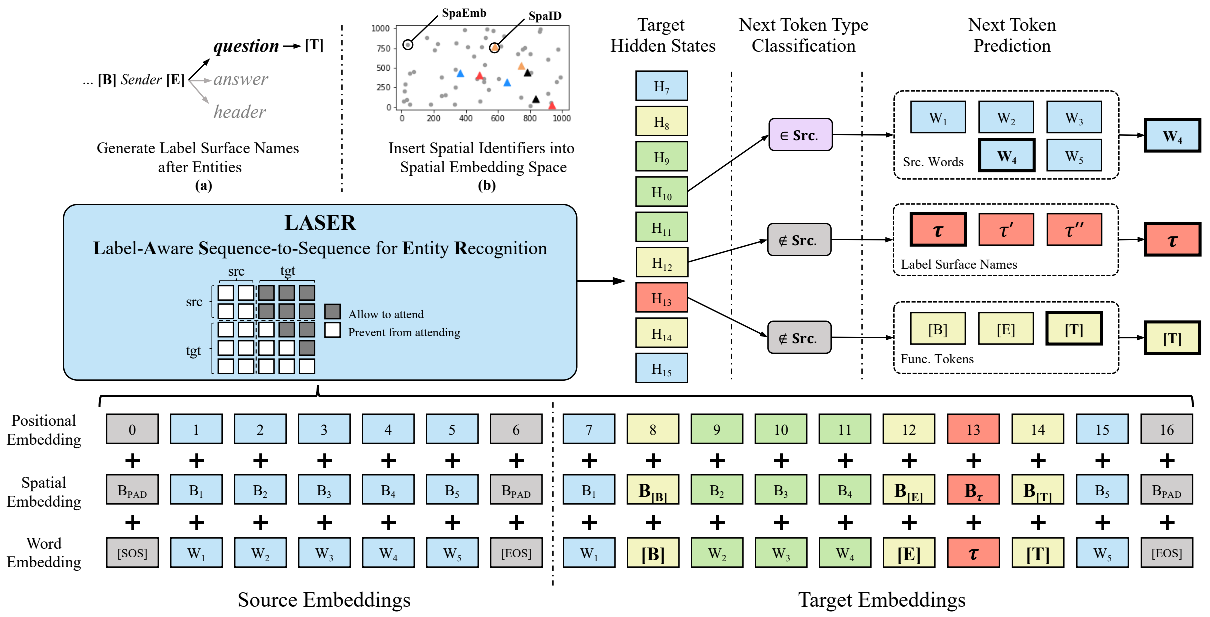


Figure 1: The Framework of LASER: [B], [E], [T] denote the boundaries; τ, τ', τ'' are the label surface names; (a) is the process of generative labeling scheme; (b) shows the alignment of the spatial identifiers and embeddings.

4 Our LASER Framework

In this section, we introduce our label-aware sequence-to-sequence framework for entity recognition in document images. First, we introduce our method in a bird's eye view. Then we dive into the details of each part including the multi-modal prefix language model, the label-aware generation.

4.1 Overview

Our proposed LASER is a label-aware sequence-to-sequence model for entity recognition in document images. The framework is shown in Figure 1. The model follows the prefix language model paradigm (Raffel et al., 2019; Dong et al., 2019; Bao et al., 2020) and is built upon the pre-trained language model, LayoutReader (Wang et al., 2021). With extensive knowledge learned in pre-training stage, the model leverages the semantic meaning of label surface names during generation.

Since the functional tokens (e.g. [B], [E]) and the label surface names are foreign words in the given page, their layout features are nonexistent. We use trainable vectors as special layout identifiers for these extra tokens and these vectors are well aligned into the spatial embedding space. In this way, the spatial correspondence between layout formats and labels can be learned.

To reinforce the model to distinguish the functional tokens (e.g. [B], [E]) and ordinary words, an extra binary classification module is added, and the probability is used in the next token prediction.

Equipped with all the components, our proposed model is able to conduct entity recognition efficiently and effectively under the few-shot setting.

4.2 Multi-modal Prefix LM

LASER is built on the layout-aware prefix language model, LayoutReader (Wang et al., 2021). Prefix language model refers to a multi-layered Transformer where the source sequence and target sequence are packed together and a "partially-triangle" mask is used to control the attention between tokens in the two sequences. In LASER, the source sequence has full self-attention and the target sequence only attends to the previous tokens so the conditional generative probability is learned.

Input Embedding The input embedding layer of LASER includes the word embedding, spatial embedding, and positional embedding. We normalize and round the bounding box coordinates to integers ranging from 0 to 1000, and embed them as trainable vectors as spatial embeddings (Xu et al., 2020, 2021a,b; Wang et al., 2021). So the input embeddings of the ordinary words are as follows:

$$e_{w_i} = \text{WordEmb}(w_i) + \text{SpaEmb}(B_i) + \text{PosEmb}(i)$$

where WordEmb, SpaEmb, PosEmb are the word embedding, the spatial embedding, and the positional embedding lookup tables, respectively; i is the index of the word in the packed sequence.

The functional tokens and label surface names are new tokens in the given page. We cannot extract the layout features from the bounding boxes of them because their bounding boxes are nonexistent. Instead of the actual bounding boxes, we design unique embedding vectors for each new tokens as their layout identifiers. These identifiers can perform in the same way as real bounding boxes during training because these identifiers are the vectors of the same dimension of the spatial embeddings and added to the input embedding. The input embedding replaces the spatial embedding with the spatial identifiers:

$$e_\lambda = \text{WordEmb}(\lambda) + \text{SpaID}(\lambda) + \text{PosEmb}(i)$$

where SpaID is the spatial identifier lookup table; i is the index of the word in the packed sequence; $\lambda \in \{\texttt{[B]}, \texttt{[E]}, \texttt{[T]}, \tau_1, ..., \tau_t\}$.

Within the input embedding layer, the pre-trained model learns the semantic and layout formats from word embeddings or spatial features. The spatial embeddings are already pre-trained and further fine-tuned in the downstream tasks, and the spatial identifiers are new to the model and completely trained in the downstream tasks.

Attention Mask As mentioned, LASER depends on a "partially-triangle" mask to realize sequence-to-sequence training within one encoder. To be more specific, the "partially-triangle" attention mask has two parts, the source part and the target part. In the source part, the tokens can attend to each other, which enables the model to be aware of the entire sequence. In the target part, to predict the next token in a sequence-to-sequence way, we design the "triangle" mask which prevents the tokens from attending to the tokens after them. Therefore, the generative probability conditioned on the previous tokens can be computed.

Output Hidden States To learn the conditional generative probability of the next token, we take the output hidden states corresponding to the target sequence which is denoted as $h_{n+1}, h_{n+2}, ..., h_{n+m}$, where $n+1$ is the beginning of the target sequence in the packed sequence. According to the "partially-triangle" attention mask, h_{n+k} is produced with the attention to the source tokens and the previous target tokens, i.e., the input embeddings whose index ranges from 1 to $n+k$. Therefore, h_{n+k} is used to predict the $(k+1)$-th token in the target sequence.

4.3 Label-aware Generation

In the sequence-to-sequence setting, LASER estimates the probability of next token conditioned on the previous context, i.e. $P(x_k|x_{<k})$ and $x_k \in \mathcal{C}$, where $\mathcal{C} = \{w_1...w_n\} \cup \{\tau_1...\tau_t\} \cup \{\texttt{[B]}, \texttt{[E]}, \texttt{[T]}\}$ is the set of all candidate words. Following LayoutReader, we restrain the candidates within the source words instead of the whole dictionary, and we go beyond it and extend the candidate set to include the functional tokens and label surface names. Moreover, to distinguish whether the next word belongs to the source or not, we design an extra binary classification module.

Specifically, we take the hidden states h_k to predict whether the next token is from the source or not. We denote the probability $P(x_{k+1} \in \text{src}) = p_{k+1}$. Then we use p_{k+1} to weight the next token prediction. The probability that the next token is the i-th word in the source is computed as follows:

$$P(x_{k+1} = w_i|x_{\leq k}) = \frac{p_{k+1}\exp\left(e_{w_i}^T h_k + b_k\right)}{\sum_j \exp\left(e_{w_j}^T h_k + b_k\right)}$$

where w_i is the i-th word in the source; e_{w_i} is the input embedding of w_i; b_k is the bias.

Similarly, the probability that the next token is one of the functional tokens or label surface names is computed as follows:

$$P(x_{k+1} = \lambda|x_{\leq k}) = \frac{(1 - p_{k+1})\exp\left(e_\lambda^T h_k + b'_k\right)}{\sum_{\lambda'} \exp\left(e_{\lambda'}^T h_k + b'_k\right)}$$

where λ is a functional token or label surface name, i.e. $\lambda \in \{\texttt{[B]}, \texttt{[E]}, \texttt{[T]}, \tau_1, ..., \tau_t\}$; $1 - p_{k+1}$ is the probability that $(k+1)$-th token is a functional token or label surface name; b'_k is the bias.

Label Semantics Learning With the log likelihood loss of generative language modeling, the model maximize the dot production between the hidden states h and the input embeddings e. The semantic correlation is learned considering that the input embeddings of the labels surface names are encoded in the word embeddings.

Spatial Identifier Learning From the layout format perspective, the input embedding of the label surface names also includes the spatial identifiers. When predicting the next token, the log likelihood also strengthens the relation between the spatial identifiers and the layout context. In this way, LASER inserts the spatial identifiers into the hyperspace of the spatial embeddings. In other words, LASER predicts where a certain label is more likely to be. Similar to the joint probability of language modeling, LASER maximizes the joint probability of a mixture of spatial identifiers and spatial embeddings: $P(..., B_{k-1}, B_k, \tau, B_{k+1}, ...)$ where B_k is the bounding boxes of the words in the page and the τ is the label to predict. Further visualization is conducted in Section 5.8.

4.4 Sequential Decoding

After training, LASER follows the prefix language modeling paradigm and generates the target sequence sequentially. We input the source sequence into the model and take the last hidden states to predict the first token in the target. Then we append the result to the end of input and repeatedly run the generation. We cache the states of the model and achieve generation in linear time.

5 Experiments

In this section, we conduct experiments and ablation study on FUNSD (Guillaume Jaume, 2019) and CORD-Lv1 (Park et al., 2019) under few-shot settings. We replace the original label surface names with other tokens to study the importance of semantic meaning. We also plot the heatmaps of the similarity between the layout identifiers and the spatial embeddings to interpret the spatial correspondence. Case studies are also conducted.

5.1 Experimental Setups

All the experiments are under few-shot settings using 1, 2, 4, 8 shots. We use 4 different random seeds to select the few-shot training samples and report the average performance and the standard deviation. To evaluate our model, we first convert our results into `IOBES` tagging scheme and compute the word-level precision, recall, and F-1 score using the APIs from Nakayama (2018) so that all comparisons with sequence labeling methods are under the same metrics. We believe such experiment settings guarantee the results are representative.

5.2 Datasets

Our experiments are conducted on two real-world data collections: FUNSD and CORD-Lv1. Both datasets provide rich annotations for the document image understandings includes the words and the word-level bounding boxes. The details and statistics of these two datasets are as follows.

- **FUNSD:** FUNSD consists of 199 fully-annotated, noisy-scanned forms with various appearance and format which makes the form understanding task more challenging. The word spans in this datasets are annotated with three different labels: `header`, `question` and `answer`, and the rest words are annotated as `other`. We use the original label names.
- **CORD-Lv1:** CORD consists of about 1000 receipts with annotations of bounding boxes and textual contents. The annotated entities have mulit-level labels. We select the first level and denote the dataset as CORD-Lv1. The first level labels include `menu`, `void-menu`, `subtotal` and `total`. We simplify `void-menu` as `void` and `subtotal` as `sub`.

Table 1: Dataset Statistics

Dataset	Split	# Page	# Entity/Page
FUNSD	Train	149	49.74
	Test	50	46.64
CORD-Lv1	Train	800	13.88
	Test	100	13.36

5.3 Compared Methods

We evaluate LASER against several strong sequence labeling methods as follows.

- **BERT** (Devlin et al., 2018) is a text-only auto-encoding pre-trained language model using the large-scale mask language modeling. We fine-tune the pre-trained BERT-base model with the few-shot training samples on each datasets.
- **RoBERTa** (Liu et al., 2019) extends the capacity of BERT and achieves better performance in multiple natural language understanding tasks. We also conduct the fine-tuning with few-shot training samples.
- **LayoutLM** (Xu et al., 2020) is a multi-modal language model which includes the layout and text information. It is built upon BERT and adds the extra spatial embeddings into the BERT embedding layer. Following LayoutLM, LayoutLMv2 (Xu et al., 2021a) leverages extra computer vision features and improves the performance, which are strong signals but absent in our settings. For a fair comparison, we do not include LayoutLMv2 in our comparative experiments.

These compared methods are in their base version and follow the `IOBES` tagging scheme.

We denote our model as LASER. In order to understand how much the label surface names can tell, we introduce an ablation version LASER (IRLVT) by replacing the label surface names with irrele-

Table 2: Evaluation Results with Different Sizes of Few-shot Training Samples: **Bold** denotes the best model; Underline denotes the second-best model.

$\lvert\mathcal{P}\rvert$	Model	FUNSD			CORD-Lv1		
		Precision	**Recall**	**F-1**	**Precision**	**Recall**	**F-1**
1	BERT	13.68±4.16	23.24±6.97	15.90±2.67	27.16±2.15	38.53±3.61	31.85±2.67
	RoBERTa	11.15±4.11	21.96±7.51	14.00±3.90	24.70±2.31	32.67±2.57	28.09±2.18
	LayoutLM	12.19±1.29	23.26±10.45	14.31±3.67	32.24±2.20	**47.10±3.03**	38.27±2.52
	LASER	**37.63±1.61**	**29.53±9.08**	**32.53±5.57**	**41.70±6.51**	37.02±6.30	**39.17±6.30**
	(IRLVT)	35.48±3.17	23.14±8.14	27.54±6.70	43.27±5.06	36.35±5.48	39.48±5.49
	(w/o SpaID)	33.74±7.40	16.21±15.41	19.84±14.15	43.63±3.49	34.24±2.27	38.34±2.54
2	BERT	15.00±3.24	30.08±5.22	19.57±2.50	35.76±6.54	51.59±8.01	42.22±7.23
	RoBERTa	12.75±2.08	24.87±6.29	16.58±2.50	35.42±5.76	49.29±9.80	41.19±7.33
	LayoutLM	16.11±3.71	31.64±11.71	20.97±5.87	45.68±6.77	**62.91±5.98**	52.88±6.64
	LASER	**37.16±5.59**	**39.85±10.18**	**38.35±7.75**	**55.39±4.94**	54.21±5.89	**54.79±5.42**
	(IRLVT)	38.03±4.49	38.89±9.73	38.29±7.09	53.91±5.18	51.73±6.10	52.79±5.66
	(w/o SpaID)	35.91±7.44	30.63±14.73	32.41±11.69	54.19±4.88	51.64±6.23	52.87±5.57
4	BERT	18.93±1.28	38.24±4.31	25.22±1.42	42.49±10.11	59.36±11.34	49.48±10.78
	RoBERTa	18.30±2.80	34.15±6.63	23.56±3.11	46.62±13.39	59.23±16.53	52.15±14.74
	LayoutLM	28.12±2.86	48.89±7.75	35.61±4.10	49.50±10.66	**66.28±9.38**	56.58±10.40
	LASER	**43.03±4.39**	**50.74±7.57**	**46.54±5.77**	**61.89±8.74**	61.41±9.72	**61.64±9.22**
	(IRLVT)	42.91±3.69	50.02±7.43	46.14±5.35	62.14±8.34	60.94±10.25	61.51±9.33
	(w/o SpaID)	42.61±3.38	46.73±7.07	44.51±5.06	62.00±9.88	60.82±11.83	61.37±10.84
8	BERT	28.09±5.32	45.59±6.58	34.71±5.84	54.88±6.06	71.11±3.48	61.88±5.19
	RoBERTa	27.79±3.49	42.23±6.62	33.51±4.60	57.16±6.05	70.88±4.13	63.24±5.37
	LayoutLM	46.95±3.48	**64.42±5.02**	54.31±4.05	62.60±5.67	**77.10±3.20**	**69.05±4.77**
	LASER	**50.66±2.48**	61.34±3.13	**55.49±2.76**	**68.70±7.03**	68.30±7.59	68.50±7.31
	(IRLVT)	49.03±3.09	59.62±3.83	53.81±3.40	67.84±6.04	66.98±6.69	67.40±6.36
	(w/o SpaID)	52.03±1.81	62.59±3.05	56.82±2.33	67.06±5.99	66.61±7.02	66.83±6.49

vant tokens, which by default are [w, x, y, z]. We also remove the spatial identifiers of label surface names and compare the performance of LASER (w/o SpaID) to study the layout format learning.

5.4 Implementation Details

We build LASER on the base of LayoutReader. We use the Transformers (Wolf et al., 2019) and the s2s-ft toolkits from the repository of Dong et al. (2019). We use one NVIDIA A6000 to finetune with batch size of 8. We optimize the model with AdamW optimizer and the learning rate is 5×10^{-5}.

5.5 Experimental Results

From Table 2, the results show that, under few-shot settings, our proposed generative labeling models achieves the SOTA overall performance compared with sequence labeling models. Specifically, compared with the second-best baseline, LASER improves the F-1 scores by 11.53% on FUNSD and by 1.58% on CORD-Lv1 on average across the different shots and LASER (IRLVT) and (w/o SpaID) also surpasses the baselines under most settings.

Moreover, the improvement on precision is remarkable. LASER improves the precision by 15.91% on FUNSD and by 9.42% on CORD-Lv1 on average across the different shots. Especially, under 1-shot setting, it surpasses the best sequence labeling model on FUNSD by 23.95% on precision, 6.27% on recall and 16.63% on F-1 score.

We can also observe a drop in the improvement with the increasing number of training samples. We conclude that, with enough training samples, the sequence labeling learns the meaning of each label and the semantics of each label surface names no longer provides extra useful information.

5.6 Ablation Study

From Table 2, we compare the performance of LASER and LASER (IRLVT) and conclude that the quality of label surface names is of great importance to the performance, which means if the label surface names are not well designed and not quite related to the entities, then the semantic correlation is useless to provide semantic signals and sometimes even harm the performance. This explains why there is about 0.3% drop in the F-1 score of CORD-Lv1 under the 1-shot setting.

Comparing the performance of LASER and LASER (w/o SpaID), the spatial correspondence learned through the spatial identifiers is helpful to

Table 3: Semantic Correspondence Study: IRLVT uses the irrelevant tokens, [*w*, *x*, *y*, *z*], as labels; ORIG uses the original label surface names; Sub1 uses [*info*, *etc*, *small*, *number*] as substitutes; Sub2 uses [*page*, *non*, *part*, *price*] as substitutes. **Bold** denotes the best model; Underline denotes the second-best model.

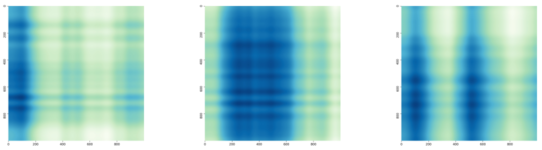
$\|\mathcal{P}\|$	Label	CORD-Lv1		
		Precision	**Recall**	**F-1**
1	IRLVT	**43.27±5.06**	36.35±5.84	39.48±5.49
	ORIG	41.70±6.51	37.02±6.30	39.17±6.30
	Sub1	42.50±6.47	36.80±6.78	39.43±6.64
	Sub2	43.21±7.16	**38.41±5.79**	**40.63±6.32**
2	IRLVT	53.91±5.18	51.73±6.10	52.79±5.66
	ORIG	55.39±4.94	54.21±5.89	54.79±5.42
	Sub1	55.54±5.83	52.71±4.97	54.07±5.35
	Sub2	**56.12±4.70**	**54.27±5.96**	**55.16±5.34**
4	IRLVT	62.14±8.34	60.94±10.25	61.51±9.33
	ORIG	61.89±8.74	61.41±9.72	61.64±9.22
	Sub1	62.23±9.45	61.52±10.27	61.87±9.86
	Sub2	**63.14±9.59**	**62.21±11.62**	**62.64±10.60**
8	IRLVT	67.84±6.04	66.98±6.69	67.40±6.36
	ORIG	68.70±7.03	68.30±7.59	68.50±7.31
	Sub1	**69.77±7.46**	**68.69±7.75**	**69.23±7.60**
	Sub2	68.89±6.98	67.92±8.03	68.39±7.48

the performance under most settings. However, there is a drop in the 8-shot setting on FUNSD. We conclude that, distinct from the simpler formats of receipts in CORD-Lv1, the pages in FUNSD have various formats making it harder to well align the spatial identifiers into the spatial embedding space.

5.7 Label Surface Name Choices

In the experiment of few-shot entity recognition, we compare LASER with LASER (IRLVT). To further study the role of label surface names, we try more different sets of words as substitutes and compare the results. The label surface names of FUNSD already provide remarkable improvements, so we only conduct the semantic correspondence study on CORD-Lv1.

From Table 3, we observe that LASER with original labels performs better than the one with irrelevant label tokens, and when we replace the original labels with more relevant tokens, such as Sub2, there is further improvements. We attribute it to the stronger semantic correspondence that can be learned by the generative labeling scheme of LASER. We also conclude that the semantic meanings of the label surface names are useful to bridge the gap between the labels and entities.



(a) Header (b) Question (c) Answer

Figure 2: Spatial correspondence visualization on FUNSD for different entity types.

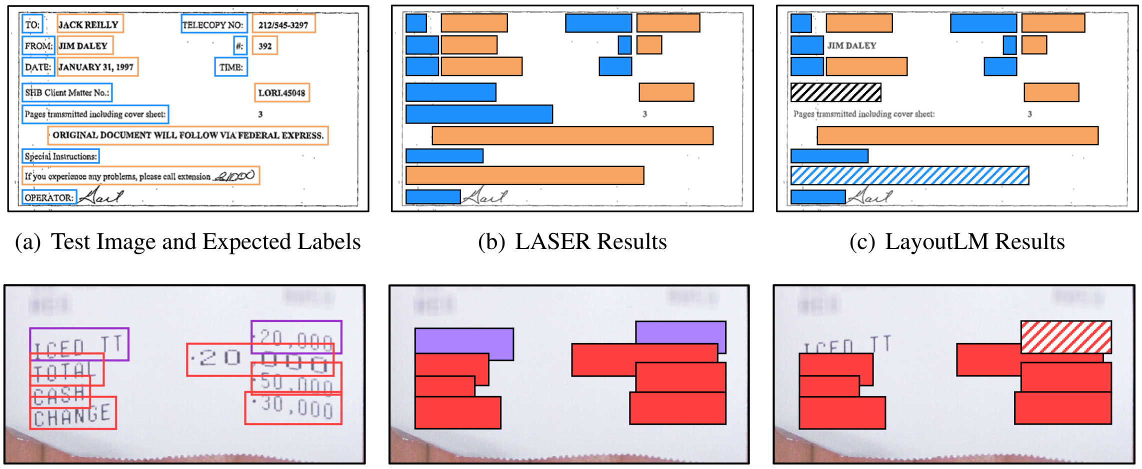
5.8 Spatial Correspondence Interpretation

In this section, we study the ability of LASER to capture the spatial correspondence between certain areas and the labels. The experiment is based on the results of LASER on FUNSD with 4 shots. As mentioned in Section 4.2, we design unique spatial identifiers for the label surface names. The identifiers are in the same form as the spatial embeddings and LASER inserts the identifiers into the original spatial embedding space during sequence-to-sequence training. Ideally, the model can learn where a certain label is more likely to appear. To visualize such patterns, we compute the cosine similarity matrix M of identifiers and the spatial embeddings as $M_{ij} = \cos\left(\text{SpaID}(\tau), \text{SpaEmb}(\text{(i, j)})\right)$ where (i, j) is the normalized coordinate pair; $\tau \in \{\tau_1, ..., \tau_t\}$. Then we plot the heatmap of the similarity matrix, where the highlight areas mean the higher similarities.

From Figure 2, we observe that the label `header` is more likely to be in the middle column of the page and may appear in the bottom part as well when there are multiple paragraphs. Intuitively, the label `question` and `answer` should appear in pairs and it is observed in Figure 2 that their heatmaps are almost complementary to each other. Several examples from FUNSD are selected to demonstrate the visualization results in Appendix. Comparing the examples and the visualization results, we conclude that the spatial identifiers of labels capture the formats of pages and LASER leverages these features to better extract the entities under few shot settings.

5.9 Case Study

We visualize cases from the 8-shot setting. From Figure 3, we observe LASER can extract the entities correctly, and the errors of LayoutLM comes from the failure to extract the entities or wrong entity type predictions. Since the sequence labeling groups the words into spans through `IOBES` tagging, which creates great uncertainty. Meanwhile,



(a) Test Image and Expected Labels (b) LASER Results (c) LayoutLM Results

(d) Test Image and Expected Labels (e) LASER Results (f) LayoutLM Results

Figure 3: Case Studies. (a), (b), (c) are from FUNSD; (d), (e), (f) are from CORD-Lv1; 🟦, 🟧, ⬛ denote `question`, `answer`, `other`; 🟪, 🟥 denote `menu`, `total`; □, ▨ denote the right, wrong predictions.

LASER also learns `questions` and `answers` appear in pairs (see Figure 3(b)). It also properly predicts a numerical string as `menu` even if numbers are likely to be `total` (see Figure 3(e)).

6 Related Work

Layout-aware LMs. Since the post-OCR processing has great application prospects, existing works propose to adapt the language pre-training to the layout formats learning. LayoutLM (Xu et al., 2020) is the pioneer in this area, which successfully uses the coordinates to represent the layout information in the embedding layer of BERT (Devlin et al., 2018). Following LayoutLM, the upgraded version, LayoutLMv2 (Xu et al., 2021a), is further proposed to leverage the visual features and benefits from the alignment between words and the regions in the page. LAMBERT (Garncarek et al., 2021) and BROS (Hong et al., 2020) continue studying the layout representation which uses the sinusoidal function or apply the relative positional biases from T5 (Raffel et al., 2019).

Generalized Seq2Seq. Sequence-to-sequence architecture is basic in natural language processing and is originally designed for machine translation. With the rise of large pre-trained models, sequence-to-sequence models are increasingly used with new problem formulation. Existing works exploit the potential latent knowledge and stronger representation ability of sequence-to-sequence modeling. GENRE (De Cao et al., 2020) creatively reformulates the entity retrieval task into the sequence-to-sequence settings. It inferences the lined entities using the generation of BART. Recent works on prompt learning also leverage the pre-trained sequence-to-sequence language models to conduct few shot learning (Liu et al., 2021; Puri and Catanzaro, 2019; Hambardzumyan et al., 2021).

7 Conclusions and Future Work

In this paper, we present LASER, a label-aware sequence-to-sequence framework for entity recognition in document images under few-shot settings. It benefits from the generative labeling scheme which reformulates the entity recognition task into the sequence-to-sequence setting. The label surface names are embedded into the generated sequence. Compared with the sequence labeling methods, LASER leverages the rich semantics of the label surface names and overcome the limitation of training data. Moreover, we design spatial identifiers for each label and well insert them into the spatial embedding hyperspace. In this way, LASER can inference the entity labels from the layout formats perspective and empirical experiments demonstrate our method can learn the layout formats though limited number of training samples.

For further research, we will investigate the selection of label surface names and how to better leverage the semantics from the pre-trained sequence-to-sequence models. We also notice that such labeling scheme can cope with unknown categories. We will focus on the generalization of our method. Meanwhile, our method is not constrained in the scenario of document images, and we will apply it to general text-only entity recognition tasks.

Ethical Consideration

This paper focuses on the entity recognition in document images under few-shot setting. Our architecture are built upon open-source models and all the datasets are available online. We will release the code and datasets on https://github.com/anonymous. Therefore, we do not anticipate any major ethical concerns.

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2020. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Graliński. 2021. Lambert: Layout-aware language modeling for information extraction. In *International Conference on Document Analysis and Recognition*, pages 532–547. Springer.

Jean-Philippe Thiran Guillaume Jaume, Hazim Kemal Ekenel. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *Accepted to ICDAR-OST*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*.

Teakgyu Hong, DongHyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2020. Bros: A pre-trained language model for understanding texts in document.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lluis Marquez, Pere Comas, Jesús Giménez, and Neus Catala. 2005. Semantic role labeling as sequential tagging. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 193–196.

Hiroki Nakayama. 2018. seqeval: A python framework for sequence labeling evaluation. Software available from https://github.com/chakki-works/seqeval.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155.

Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. 2021. Layoutreader: Pre-training of text and layout for reading order detection.

Zilong Wang, Mingjie Zhan, Xuebo Liu, and Ding Liang. 2020. Docstruct: A multimodal method to extract hierarchy structure in document for general form understanding. *arXiv preprint arXiv:2010.11685*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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. 2021a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL) 2021*.

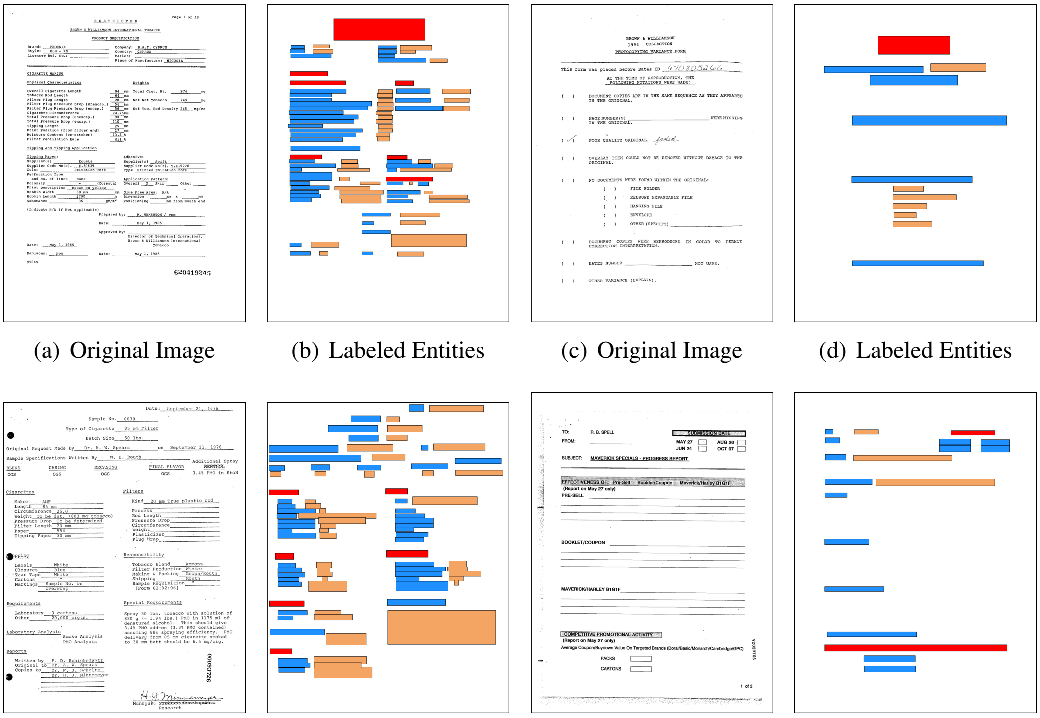
Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021b. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding.

Appendix

A Layout Format Examples

Several examples are listed in Figure 4. We observe that `questions` and `answers` are roughly organized in columns and appear in pairs. Most `headers` are located in the upper part of each page but there are also cases where `header` appear in the bottom of the page. There patterns align with the visualization results in Section 5.8.



(a) Original Image (b) Labeled Entities (c) Original Image (d) Labeled Entities

(e) Original Image (f) Labeled Entities (g) Original Image (h) Labeled Entities

Figure 4: Layout Format Examples from FUNSD: 🟦, 🟧, 🟥 denotes `question`, `answer`, `header`.