

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

Zilong Wang

University of California, San Diego
zlwang@ucsd.edu

Jingbo Shang*

University of California, San Diego
jshang@ucsd.edu

Abstract

Entity recognition is a fundamental task in understanding document images. Traditional sequence labeling frameworks treat the entity types as class IDs and rely on extensive data and high-quality annotations to learn semantics which are typically expensive in practice. In this paper, we aim to build an entity recognition model requiring 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 type semantics and also embed the labels into the spatial embedding space to capture the spatial correspondence between regions and labels. Specifically, we go beyond sequence labeling and develop a novel label-aware seq2seq framework, LASER. The proposed model follows a new labeling scheme that generates the label surface names word-by-word explicitly after generating the entities. 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. Entity types are treated as class IDs and the semantics of the label surface names are ignored. These methods also 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 fail to learn from the data efficiently and require extensive datasets and high-quality annotations to create the connection between entities and their entity types. Meanwhile, document images typically include various formats and have a high diversity of entities within each page. It is expensive or almost impossible to enumerate all required entity types and obtain enough annotated data for them. Moreover, ethical concerns would arise when it comes to the receipts or consent forms, which makes it even harder to collect enough data.

Due to the inefficiency of traditional methods and the data limitation in real application scenarios, it is necessary to resort to few-shot learning for entity recognition in document images. We aim at exploiting the potential of a limited number of training pages and try to generalize our model on the much larger number of new pages for testing. 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 explicitly 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 and models can leverage the semantic connect between the entities and entity types.

* Jingbo Shang is the corresponding author.

To this end, we propose a label-aware sequence-to-sequence framework for entity recognition, LASER. Our implementation is based on pre-trained language model LayoutReader¹ (Wang et al., 2021), which is a layout-aware pre-trained sequence-to-sequence model.

As shown in Figure 1, LASER extends the architecture of LayoutReader for our proposed generative labeling scheme to better solve the few-shot entity recognition task for document images. Specifically, after generating certain word spans, the model can choose to generate either the following words in the source sequence or label surface names. The entity labels are explicitly inserted in the generated sequence so that the probability of the entity types conditioned on the entity, $P(\text{type}|\text{entity})$, can be maximized not only by the signals from the training data but also by the knowledge from the pre-training of the language models. We also embed the label surface names into the spatial embedding space, so the generation of labels is also aware of the correlation between labels and the regions in the page.

Benefit from the novel generative labeling scheme and the semantics of labels, LASER is able to effectively recognize entities in document images with only a limited number of training samples. In contrast, the sequence labeling models use less efficient tagging scheme, thus requiring more data and failing in the few-shot settings.

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 Github².

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) by human annotators or the OCR engines, and all the words and bounding boxes are listed in a sequence serving as the inputs from textual and layout modalities. In this way, the entities are spans of these words referring to precise concepts, which makes it possible to conduct entity recognition using sequence labeling or generative labeling scheme. We randomly select a small subset of training samples and evaluate the performance under the k -shot training, where k denotes the number of the training samples.

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 $e = [w_i, w_{i+1}, \dots, w_j]$, 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:

$$w_{i-1}, [\mathbf{B}], w_i, \dots, w_j, [\mathbf{E}], \tau_1, \dots, \tau_k, [\mathbf{T}], w_{j+1}$$

where [B] and [E] denote the start and end of the entity; $\tau_1 \dots \tau_k$ 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].

4 Our LASER Framework

In this section, we introduce our label-aware sequence-to-sequence framework for entity recog-

¹Licensed under the MIT License

²github.com/zlwang-cs/LASER-release

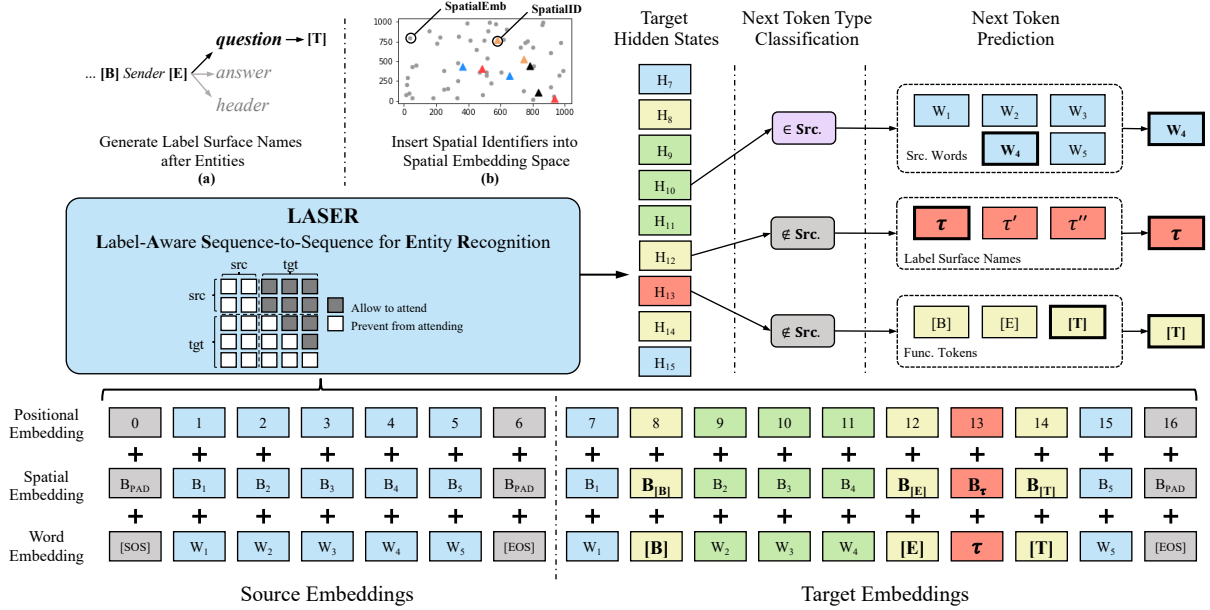


Figure 1: The Framework of LASER: [B], [E], [T] denote the boundaries; τ , τ' , τ'' are the label surface names; (a) is the process of generative labeling scheme; (b) shows the alignment of the spatial identifiers and embeddings.

tion 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., 2019a; 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 effi-

ciently 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:

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

where WordEmb, SpatialEmb, 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 to embed the functional tokens and label surface names into the spatial embedding space. The input embedding replaces the spatial embedding with the spatial identifiers:

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

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

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 $\mathbf{h}_{n+1}, \mathbf{h}_{n+2}, \dots, \mathbf{h}_{n+m}$, where $n+1$ is the beginning of the target sequence in the packed sequence. According to the “partially-triangle” attention mask, \mathbf{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, \mathbf{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 $\mathcal{C} = \{w_1 \dots w_n\} \cup \{\tau_1 \dots \tau_t\} \cup \{[\text{B}], [\text{E}], [\text{T}]\}$ 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:

$$P(x_{k+1} = w_i | x_{\leq k}) = \frac{p_{k+1} \exp(\mathbf{e}_{w_i}^T \mathbf{h}_k + b_k)}{\sum_j \exp(\mathbf{e}_{w_j}^T \mathbf{h}_k + b_k)}$$

where w_i is the i -th word in the source; \mathbf{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:

$$P(x_{k+1} = \lambda | x_{\leq k}) = \frac{(1 - p_{k+1}) \exp(\mathbf{e}_\lambda^T \mathbf{h}_k + b'_k)}{\sum_{\lambda'} \exp(\mathbf{e}_{\lambda'}^T \mathbf{h}_k + b'_k)}$$

where λ is a functional token or label surface name, i.e. $\lambda \in \{[\text{B}], [\text{E}], [\text{T}], \tau_1, \dots, \tau_t\}$; $1 - p_{k+1}$ 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 \mathbf{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: $P(\dots, B_{k-1}, B_k, \tau, B_{k+1}, \dots)$ 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 5.7.

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 spatial 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, 3, 4, 5, 6, 7 shots. We use 6 different random seeds to select the few-shot training samples and the data augmentation is conducted to solve the data sparsity. We train all the models using the same data and compute the average performance and the standard deviation. We only report the result of 1, 3, 5, 7 shots for space limitation. To evaluate our model, we first convert our results into IOBES tagging style 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

Table 1: Dataset Statistics.

| Dataset | # Train Pages | # Test Pages | # Entities / Page |
|----------|---------------|--------------|-------------------|
| FUNSD | 149 | 50 | 42.86 |
| CORD-Lv1 | 800 | 100 | 13.82 |

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 entities have multi-level labels. We select the first level and denote the dataset as CORD-Lv1. The first level includes `menu`, `void-menu`, `subtotal` and `total`. We simplify `subtotal` as `sub` and `void-menu` as `void`.

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.
- **LayoutReader** (Wang et al., 2021) is a layout-aware sequence-to-sequence model for reading order detection. We append a linear layer upon the hidden states to conduct sequence labeling.

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

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

| P | Model | FUNSD | | | CORD-Lv1 | | |
|---|--------------|-------------------|-------------------|-------------------|-------------------|--------------------|-------------------|
| | | Precision | Recall | F-1 | Precision | Recall | F-1 |
| 1 | BERT | 9.62±2.24 | 24.14±3.46 | 13.55±2.09 | 30.64±2.80 | 45.60±3.45 | 36.64±3.10 |
| | RoBERTa | 9.29±1.57 | 22.06±5.64 | 12.76±1.91 | 30.66±4.25 | 44.39±6.72 | 36.25±5.18 |
| | LayoutLM | <u>11.39±1.12</u> | <u>24.73±7.38</u> | <u>15.18±2.17</u> | <u>33.27±7.32</u> | 49.49±10.26 | <u>39.77±8.47</u> |
| | LayoutReader | 11.32±0.62 | 22.53±4.80 | 14.84±1.25 | 32.17±4.64 | 45.61±6.54 | 37.70±5.31 |
| | LASER | 30.40±4.89 | 35.20±7.20 | 32.36±5.14 | 47.63±3.90 | 45.52±5.84 | 46.24±3.01 |
| 3 | BERT | 16.42±4.30 | 34.74±5.36 | 22.19±5.05 | 39.62±3.99 | 56.65±4.03 | 46.58±3.94 |
| | RoBERTa | 16.71±3.63 | 31.28±3.55 | 21.66±3.84 | 44.51±4.69 | 60.18±4.69 | 51.15±4.70 |
| | LayoutLM | <u>28.67±6.56</u> | 47.22±8.31 | <u>35.42±7.00</u> | <u>47.68±7.49</u> | 63.93±7.04 | <u>54.57±7.46</u> |
| | LayoutReader | 22.37±2.03 | 35.19±4.97 | 27.19±2.56 | 43.85±4.72 | 56.90±2.47 | 49.47±3.95 |
| | LASER | 43.66±1.97 | <u>47.08±5.72</u> | 45.21±3.74 | 61.16±3.11 | <u>60.33±5.65</u> | 60.63±4.00 |
| 5 | BERT | 20.57±2.59 | 39.25±1.10 | 26.93±2.46 | 45.73±4.31 | 63.29±3.68 | 53.06±4.14 |
| | RoBERTa | 19.47±2.32 | 35.04±1.89 | 24.94±1.93 | 52.21±4.55 | <u>66.63±5.52</u> | 58.54±4.92 |
| | LayoutLM | 39.24±4.33 | 58.20±2.45 | 46.72±3.12 | <u>56.13±7.39</u> | 71.66±6.13 | <u>62.91±7.04</u> |
| | LayoutReader | 27.52±3.44 | 41.17±4.01 | 32.89±3.28 | 51.97±8.42 | 63.82±7.87 | 57.24±8.32 |
| | LASER | 47.25±1.93 | <u>52.85±1.22</u> | 49.87±1.29 | 65.62±3.79 | 64.90±5.78 | 65.23±4.70 |
| 7 | BERT | 21.44±2.07 | 40.87±3.79 | 28.09±2.48 | 50.13±4.35 | 66.67±3.67 | 57.20±4.07 |
| | RoBERTa | 23.68±3.06 | 38.74±3.54 | 29.32±3.08 | 55.14±4.49 | 69.35±4.16 | 61.43±4.42 |
| | LayoutLM | 43.23±5.27 | 61.73±5.97 | 50.76±5.30 | <u>62.87±3.98</u> | 76.38±2.72 | 68.96±3.49 |
| | LayoutReader | 31.22±3.14 | 45.08±3.83 | 36.85±3.26 | 54.43±5.89 | 65.48±5.34 | 59.42±5.68 |
| | LASER | 50.62±3.26 | <u>53.63±2.89</u> | 51.98±2.00 | 68.02±3.16 | 66.87±4.82 | <u>67.40±3.76</u> |

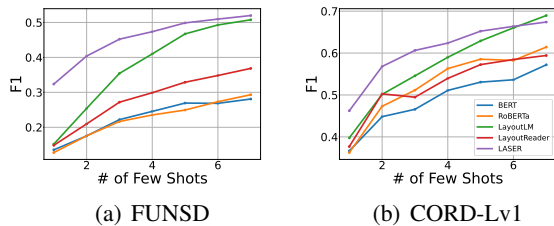


Figure 2: F-1 Curves with Different Sizes of Few-shot Training Samples.

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. (2019a). 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 and Figure 2, the results show that, under few-shot settings, our proposed model, LASER, achieves the SOTA overall performance compared with sequence labeling models. We conclude that the gain of performance comes mostly from the generative labeling scheme since LASER largely outperforms LayoutReader although both of them share the same backbone.

Specifically, compared with the second-best

baseline, LASER improves the F-1 scores by 8.59% on FUNSD and by 3.32% on CORD-Lv1 on average across the different shots and LASER (IRLVT) also surpasses the baselines under most settings.

Moreover, the improvement on precision is remarkable. LASER improves the precision by 12.35% on FUNSD and by 10.62% on CORD-Lv1 on average across the different shots. Especially, under 1-shot setting, it surpasses the best sequence labeling model on FUNSD by 19.01% on precision, 10.47% on recall and 17.18% 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.

Based on these comparison, we safely come to the conclusion that our proposed generative labeling scheme is superior to the traditional sequence labeling scheme in few shot settings.

5.6 Ablation Study

In the ablation study, we aim at study the role of the label surface names. We introduce an ablation version, LASER (IRLVT), by replacing the label surface names with irrelevant tokens. We also design more different sets of words as substitutes denoted Sub1 and Sub2. The detailed substitutes

Table 3: Ablation Study of Different Label Surface Names in LASER. IRLVT uses the irrelevant tokens as labels; ORIG uses the original label surface names; Sub1 and Sub2 use some reasonable alternative label surface names. as substitutes. **Bold** denotes the best model; Underline denotes the second-best model.

| P | FUNSD | | | | CORD-Lv1 | | | |
|---|---------------------------------|-------------------|-------------------|-------------------|---------------------------------|-------------------|-------------------|-------------------|
| | Label Surface Names | Precision | Recall | F-1 | Label Surface Names | Precision | Recall | F-1 |
| 1 | IRLVT [x, y, z] | 30.64±5.89 | 33.45±9.14 | 31.62±6.61 | IRLVT [w, x, y, z] | 48.57±4.93 | 44.12±6.36 | 45.84±3.57 |
| | ORIG [header, question, answer] | 30.40±4.89 | <u>35.20±7.20</u> | 32.36±5.14 | ORIG [menu, void, sub, total] | 47.63±3.90 | <u>45.52±5.84</u> | <u>46.24±3.01</u> |
| | Sub1 [title, key, value] | 31.78±4.75 | 34.21±7.44 | <u>32.66±5.10</u> | Sub1 [info, etc, small, number] | 48.12±4.15 | 48.47±6.60 | 48.04±4.06 |
| | Sub2 [page, topic, value] | 30.90±5.20 | 35.97±8.57 | 33.03±6.31 | Sub2 [page, non, part, price] | 45.59±5.68 | 44.09±7.87 | 44.38±5.39 |
| 3 | IRLVT [x, y, z] | 43.51±1.46 | 47.92±5.93 | 45.44±3.36 | IRLVT [w, x, y, z] | 61.50±2.52 | 59.17±4.11 | 60.27±2.99 |
| | ORIG [header, question, answer] | 43.66±1.97 | 47.08±5.72 | 45.21±3.74 | ORIG [menu, void, sub, total] | 61.16±3.11 | 60.33±5.65 | <u>60.63±4.00</u> |
| | Sub1 [title, key, value] | 43.87±1.33 | 47.11±6.07 | 45.26±3.44 | Sub1 [info, etc, small, number] | 61.54±2.76 | 58.79±6.76 | 60.00±4.57 |
| | Sub2 [page, topic, value] | 43.88±1.34 | 48.01±6.86 | 45.65±3.93 | Sub2 [page, non, part, price] | 61.85±2.16 | <u>60.29±2.85</u> | 61.03±2.10 |
| 5 | IRLVT [x, y, z] | 46.94±1.87 | 52.96±2.03 | 49.74±1.63 | IRLVT [w, x, y, z] | 63.67±3.82 | 61.10±5.21 | 62.33±4.48 |
| | ORIG [header, question, answer] | 47.25±1.93 | 52.85±1.22 | 49.87±1.29 | ORIG [menu, void, sub, total] | 65.62±3.79 | 64.90±5.78 | 65.23±4.70 |
| | Sub1 [title, key, value] | 47.43±2.29 | 52.19±2.09 | 49.68±1.98 | Sub1 [info, etc, small, number] | 65.05±5.59 | 63.64±7.16 | 64.31±6.34 |
| | Sub2 [page, topic, value] | 47.46±2.18 | 53.50±1.01 | 50.26±1.16 | Sub2 [page, non, part, price] | 65.57±3.04 | 64.71±3.97 | <u>65.12±3.38</u> |
| 7 | IRLVT [x, y, z] | 50.30±2.26 | 54.14±3.48 | <u>52.08±2.26</u> | IRLVT [w, x, y, z] | 66.08±3.26 | 64.73±5.08 | 65.32±3.74 |
| | ORIG [header, question, answer] | 50.62±3.26 | 53.63±2.89 | 51.98±2.00 | ORIG [menu, void, sub, total] | 68.02±3.16 | 66.87±4.82 | 67.40±3.76 |
| | Sub1 [title, key, value] | 50.22±3.20 | 53.79±3.13 | 51.88±2.56 | Sub1 [info, etc, small, number] | 67.61±4.19 | 66.64±5.72 | 67.08±4.72 |
| | Sub2 [page, topic, value] | <u>50.43±2.88</u> | <u>54.03±2.71</u> | 52.10±2.09 | Sub2 [page, non, part, price] | 66.64±3.97 | 63.59±7.00 | 65.02±5.47 |

are introduced in Table 3.

To implement the ablation study, we simply replace the word embedding of label surface names. For example, in LASER (Sub1) on FUNSD, we use the wording embedding of *title* instead of the original *header*.

From Table 3, we compare the performance of all the ablation models. We observe that LASER performs differently with distinct label semantics. In most cases, the human-designed labels can provide stronger semantic correlation with the entities than the irrelevant labels so they can further improve the performance. However, there are also drops due to improper labels. Overall, we conclude that the semantic meanings of the label surface names are useful to bridge the gap between the labels and entities.

5.7 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 7 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 $M_{ij} = \cos(\text{SpatialID}(\tau), \text{SpatialEmb}((i, j)))$

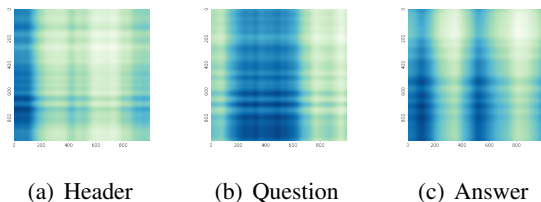


Figure 3: Spatial correspondence visualization on FUNSD for different entity types.

where (i, j) is the normalized coordinate pair; $\tau \in \{\tau_1, \dots, \tau_t\}$. Then we plot the heatmap of the similarity matrix, where the highlight areas mean the higher similarities.

From Figure 3, 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 3 that their heatmaps are almost complementary to each other. Several examples from FUNSD are selected to demonstrate the visualization results in 4. 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.

5.8 Case Study

We visualize cases from the 5-shot setting. From Figure 5, we observe LASER can extract the enti-

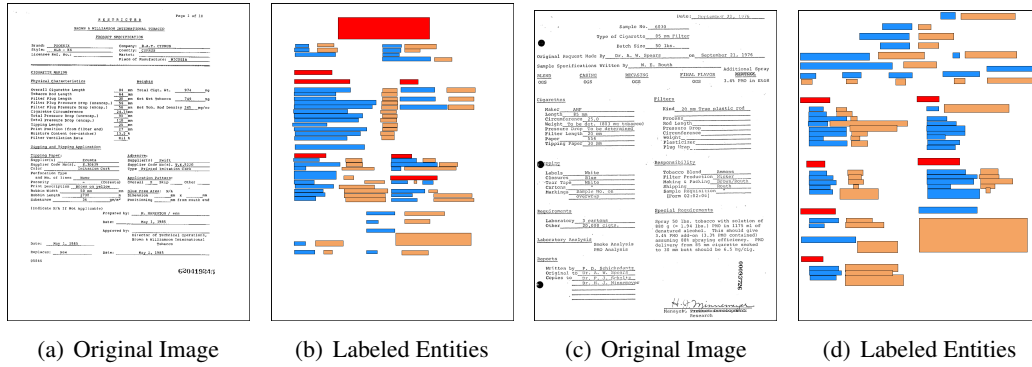


Figure 4: Layout Format Examples from FUNSD: ■, ■, ■ denotes question, answer, header.

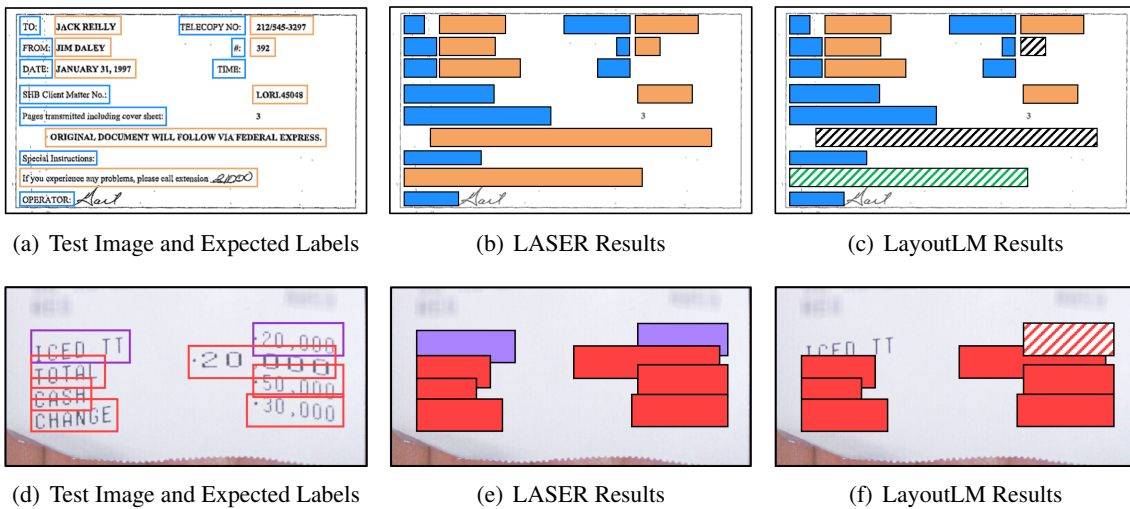


Figure 5: Case Studies. (a), (b), (c) from FUNSD; (d), (e), (f) from CORD-Lv1; ■, ■, ■, ■ denote menu, question, answer, other; ■, ■ denote menu, total; , denote the right, wrong predictions.

Table 4: Text-only Dataset Statistics

| Dataset | # Train | # Test | # Entity Type |
|-----------|---------|--------|---------------|
| OntoNotes | 60.0k | 8.3k | 18 |
| Mit Movie | 7.8k | 2.0k | 12 |

Table 5: Results of 10-way-5-shot Experiments

| Model | OntoNotes | MIT Movie |
|------------------------------|-------------------|-------------------|
| | F-1 | F-1 |
| BERT | 60.79±0.97 | 47.88±0.97 |
| RoBERTa (Huang et al., 2020) | 57.70 | 51.30 |
| UniLM | 60.82±1.26 | 51.09±1.40 |
| LASER | 61.11±1.08 | 51.88±1.27 |

ties 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, LASER also learns questions and answers appear in pairs (see Figure 5(b)). It also properly predicts a numerical string as menu even if numbers are likely to be total (see Figure 5(e)).

5.9 Text-only Entity Recognition

LASER is designed for the entity recognition task in document images where both text and layout

can be leveraged to acquire essential information. However, the generative labeling scheme is not constrained in the scenario of document images. We briefly explore the potential of the generative labeling scheme in text-only scenario. We initialize LASER with a text-only language model, UniLM (Dong et al., 2019b), based on the experiments in Wang et al. (2021), and apply it onto text-only entity recognition task. Following Huang et al. (2020), we conduct 10-way-5-shot experiments on two datasets, OntoNotes (Weischedel et al., 2013)

and MIT Movie (Liu et al., 2013), which cover general domains and review domains, respectively. The dataset statistics are shown in Table 4 and the results are as shown in Table 5. We observe that our method can also surpass the sequence labeling methods in these two datasets, showing the great potential of the generative labeling scheme in the entity recognition tasks.

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). LayoutReader (Wang et al., 2021) aims to predict the reading order of words from the OCR results. ReadingBank (Wang et al., 2021) is proposed to facilitate the pre-training of reading order detection, which annotates the reading order of millions of pages.

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 infers 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 recog-

nition 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.

Acknowledgments

We want to thank the anonymous reviewers for their insightful comments. Zilong would also like to thank the Powell Fellowship for the support during the challenging years at the beginning of his Ph.D. program.

The research was sponsored in part by National Science Foundation Convergence Accelerator under award OIA-2040727 as well as generous gifts from Google, Adobe, and Teradata. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon.

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/zlwang-cs/LASER-release>. 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. 2019a. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019b. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32.
- Łukasz Garncarek, Rafał Powalski, Tomasz Stanisławek, Bartosz Topolski, Piotr Halama, Michał Turski, and Filip Galiński. 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 Khachatryan, 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.
- Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *arXiv preprint arXiv:2012.14978*.
- Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and Jim Glass. 2013. Query understanding enhanced by hierarchical parsing structures. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 72–77. IEEE.
- 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*.
- Lluís 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*.
- Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.
- 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 All Results

All results are listed in Table 6.

Table 6: Evaluation Results with Different Sizes of Few-shot Training Samples

| \mathcal{P} | Model | FUNSD | | | CORD-Lv1 | | |
|---------------|--------------|------------|------------|------------|------------|-------------|------------|
| | | Precision | Recall | F-1 | Precision | Recall | F-1 |
| 1 | BERT | 9.62±2.24 | 24.14±3.46 | 13.55±2.09 | 30.64±2.80 | 45.60±3.45 | 36.64±3.10 |
| | RoBERTa | 9.29±1.57 | 22.06±5.64 | 12.76±1.91 | 30.66±4.25 | 44.39±6.72 | 36.25±5.18 |
| | LayoutLM | 11.39±1.12 | 24.73±7.38 | 15.18±2.17 | 33.27±7.32 | 49.49±10.26 | 39.77±8.47 |
| | LayoutReader | 11.32±0.62 | 22.53±4.80 | 14.84±1.25 | 32.17±4.64 | 45.61±6.54 | 37.70±5.31 |
| | LASER | 30.40±4.89 | 35.20±7.20 | 32.36±5.14 | 47.63±3.90 | 45.52±5.84 | 46.24±3.01 |
| | (IRLVT) | 30.64±5.89 | 33.45±9.14 | 31.62±6.61 | 48.57±4.93 | 44.12±6.36 | 45.84±3.57 |
| | (Sub1) | 31.78±4.75 | 34.21±7.44 | 32.66±5.10 | 48.12±4.15 | 48.47±6.60 | 48.04±4.06 |
| (Sub2) | 30.90±5.20 | 35.97±8.57 | 33.03±6.31 | 45.59±5.68 | 44.09±7.87 | 44.38±5.39 | |
| 2 | BERT | 12.49±3.24 | 30.01±4.55 | 17.53±3.89 | 37.66±3.79 | 55.43±3.41 | 44.82±3.78 |
| | RoBERTa | 13.12±3.08 | 27.63±5.16 | 17.56±3.59 | 40.66±3.60 | 56.60±4.87 | 47.30±4.05 |
| | LayoutLM | 19.73±5.43 | 37.32±9.58 | 25.41±6.25 | 42.92±8.27 | 60.47±8.15 | 50.15±8.44 |
| | LayoutReader | 16.66±2.58 | 29.26±5.34 | 20.96±3.01 | 44.14±7.77 | 58.50±8.32 | 50.26±8.04 |
| | LASER | 39.23±3.09 | 42.43±6.23 | 40.38±2.71 | 59.31±2.31 | 54.85±6.19 | 56.80±3.46 |
| | (IRLVT) | 38.75±3.74 | 40.37±8.17 | 39.19±4.84 | 57.05±2.87 | 53.49±7.01 | 55.01±4.19 |
| | (Sub1) | 38.47±2.95 | 41.11±7.03 | 39.43±3.83 | 57.46±4.22 | 54.72±5.87 | 55.98±4.68 |
| (Sub2) | 37.52±2.22 | 43.23±6.48 | 39.90±3.12 | 57.03±2.80 | 55.75±5.48 | 56.28±3.48 | |
| 3 | BERT | 16.42±4.30 | 34.74±5.36 | 22.19±5.05 | 39.62±3.99 | 56.65±4.03 | 46.58±3.94 |
| | RoBERTa | 16.71±3.63 | 31.28±3.55 | 21.66±3.84 | 44.51±4.69 | 60.18±4.69 | 51.15±4.70 |
| | LayoutLM | 28.67±6.56 | 47.22±8.31 | 35.42±7.00 | 47.68±7.49 | 63.93±7.04 | 54.57±7.46 |
| | LayoutReader | 22.37±2.03 | 35.19±4.97 | 27.19±2.56 | 43.85±4.72 | 56.90±2.47 | 49.47±3.95 |
| | LASER | 43.66±1.97 | 47.08±5.72 | 45.21±3.74 | 61.16±3.11 | 60.33±5.65 | 60.63±4.00 |
| | (IRLVT) | 43.51±1.46 | 47.92±5.93 | 45.44±3.36 | 61.50±2.52 | 59.17±4.11 | 60.27±2.99 |
| | (Sub1) | 43.87±1.33 | 47.11±6.07 | 45.26±3.44 | 61.54±2.76 | 58.79±6.76 | 60.00±4.57 |
| (Sub2) | 43.88±1.34 | 48.01±6.86 | 45.65±3.93 | 61.85±2.16 | 60.29±2.85 | 61.03±2.10 | |
| 4 | BERT | 18.25±3.30 | 37.90±2.93 | 24.55±3.59 | 43.94±4.14 | 61.13±4.18 | 51.09±4.12 |
| | RoBERTa | 17.99±2.84 | 34.38±4.09 | 23.52±3.07 | 49.56±4.89 | 65.19±4.85 | 56.29±4.86 |
| | LayoutLM | 33.38±3.62 | 53.71±3.24 | 41.00±2.71 | 52.15±7.90 | 68.06±6.86 | 58.99±7.66 |
| | LayoutReader | 24.61±2.58 | 38.28±3.05 | 29.89±2.51 | 48.27±9.14 | 61.29±8.49 | 53.96±9.06 |
| | LASER | 44.91±2.42 | 50.25±3.26 | 47.36±2.18 | 63.90±3.19 | 60.99±5.86 | 62.38±4.60 |
| | (IRLVT) | 46.31±1.91 | 51.74±2.55 | 48.83±1.67 | 63.68±3.72 | 60.39±7.11 | 61.93±5.50 |
| | (Sub1) | 45.58±1.63 | 51.47±2.97 | 48.29±1.65 | 63.22±3.31 | 60.39±8.23 | 61.65±5.93 |
| (Sub2) | 45.43±2.08 | 51.74±2.64 | 48.33±1.79 | 62.85±3.83 | 60.07±8.00 | 61.31±5.97 | |
| 5 | BERT | 20.57±2.59 | 39.25±1.10 | 26.93±2.46 | 45.73±4.31 | 63.29±3.68 | 53.06±4.14 |
| | RoBERTa | 19.47±2.32 | 35.04±1.89 | 24.94±1.93 | 52.21±4.55 | 66.63±5.52 | 58.54±4.92 |
| | LayoutLM | 39.24±4.33 | 58.20±2.45 | 46.72±3.12 | 56.13±7.39 | 71.66±6.13 | 62.91±7.04 |
| | LayoutReader | 27.52±3.44 | 41.17±4.01 | 32.89±3.28 | 51.97±8.42 | 63.82±7.87 | 57.24±8.32 |
| | LASER | 47.25±1.93 | 52.85±1.22 | 49.87±1.29 | 65.62±3.79 | 64.90±5.78 | 65.23±4.70 |
| | (IRLVT) | 46.94±1.87 | 52.96±2.03 | 49.74±1.63 | 63.67±3.82 | 61.10±5.21 | 62.33±4.48 |
| | (Sub1) | 47.43±2.29 | 52.19±2.09 | 49.68±1.98 | 65.05±5.59 | 63.64±7.16 | 64.31±6.34 |
| (Sub2) | 47.46±2.18 | 53.50±1.01 | 50.26±1.16 | 65.57±3.04 | 64.71±3.97 | 65.12±3.38 | |
| 6 | BERT | 20.37±2.08 | 39.58±2.90 | 26.85±2.27 | 46.39±4.56 | 63.61±4.81 | 53.62±4.65 |
| | RoBERTa | 21.70±2.37 | 37.14±1.34 | 27.32±1.96 | 51.80±5.60 | 66.54±5.87 | 58.25±5.77 |
| | LayoutLM | 41.75±3.83 | 60.47±3.23 | 49.28±3.02 | 59.31±4.23 | 74.48±3.16 | 66.02±3.86 |
| | LayoutReader | 29.19±2.10 | 43.21±2.32 | 34.82±2.09 | 52.76±4.45 | 65.62±3.47 | 58.45±3.99 |
| | LASER | 48.64±2.14 | 53.54±2.10 | 50.96±1.95 | 66.85±3.88 | 65.97±5.19 | 66.38±4.38 |
| | (IRLVT) | 48.33±2.15 | 52.68±1.56 | 50.36±1.08 | 66.01±4.07 | 63.85±5.59 | 64.87±4.63 |
| | (Sub1) | 48.49±1.96 | 52.85±1.67 | 50.53±0.99 | 65.19±4.22 | 63.37±7.10 | 64.21±5.68 |
| (Sub2) | 49.14±1.97 | 53.42±1.20 | 51.16±1.22 | 66.44±3.11 | 64.05±4.31 | 65.21±3.67 | |
| 7 | BERT | 21.44±2.07 | 40.87±3.79 | 28.09±2.48 | 50.13±4.35 | 66.67±3.67 | 57.20±4.07 |
| | RoBERTa | 23.68±3.06 | 38.74±3.54 | 29.32±3.08 | 55.14±4.49 | 69.35±4.16 | 61.43±4.42 |
| | LayoutLM | 43.23±5.27 | 61.73±5.97 | 50.76±5.30 | 62.87±3.98 | 76.38±2.72 | 68.96±3.49 |
| | LayoutReader | 31.22±3.14 | 45.08±3.83 | 36.85±3.26 | 54.43±5.89 | 65.48±5.34 | 59.42±5.68 |
| | LASER | 50.62±3.26 | 53.63±2.89 | 51.98±2.00 | 68.02±3.16 | 66.87±4.82 | 67.40±3.76 |
| | (IRLVT) | 50.30±2.26 | 54.14±3.48 | 52.08±2.26 | 66.08±3.26 | 64.73±5.08 | 65.32±3.74 |
| | (Sub1) | 50.22±3.20 | 53.79±3.13 | 51.88±2.56 | 67.61±4.19 | 66.64±5.72 | 67.08±4.72 |
| (Sub2) | 50.43±2.88 | 54.03±2.71 | 52.10±2.09 | 66.64±3.97 | 63.59±7.00 | 65.02±5.47 | |