Entity Relation Extraction as Dependency Parsing in Visually Rich Documents

Yue Zhang[1], Bo Zhang[1], Rui Wang[2]*, Junjie Cao[1], Chen Li[1], Zuyi Bao[1]

Alibaba Group, China

[1]{shiyu.zy,klayzhang.zb,puji.lc,junjie.junjiecao,zuyi.bzy}
@alibaba-inc.com
[2]mars198356@hotmail.com

Abstract

Previous works on key information extraction from visually rich documents (VRDs) mainly focus on labeling the text within each bounding box (i.e., *semantic entity*), while the relations in-between are largely unexplored. In this paper, we adapt the popular dependency parsing model, the biaffine parser, to this *entity relation extraction* task. Being different from the original dependency parsing model which recognizes dependency relations between words, we identify relations between groups of words with layout information instead. We have compared different representations of the semantic entity, different VRD encoders, and different relation decoders. For the model training, we explore multi-task learning to combine entity labeling and relation extraction tasks; and for the evaluation, we conduct experiments on different datasets with filtering and augmentation. The results demonstrate that our proposed model achieves 65.96% F1 score on the FUNSD dataset. As for the real-world application, our model has been applied to the in-house customs data, achieving reliable performance in the production setting.

1 Introduction

In real-life scenarios, there are many types of visually rich documents (VRDs), such as invoices, questionnaire forms, declaration materials and so on. These documents contain abundant layout information which helps us to understand the content while texts alone are not enough. In recent years, many works focus on how to extract key information from VRDs based on the results of OCR (Optical Character Recognition), which recognizes bounding boxes and texts within the boxes (Liu et al., 2019a; Yu et al., 2020b). Each bounding box contains 1) a group of words that belong together

from a semantic and spatial standpoint and 2) visual features such as layout, tabular structure and font size of the boxes in the document. We call such bounding boxes and texts within the boxes *semantic entities*[1], and each entity contains the word group and layout coordinates[2].

Key information extraction (KIE) is such a task to analyze visually rich documents, which usually contains two steps, *entity labeling* and *entity relation extraction*. Similar to named entity recognition (NER) and relation extraction in the traditional natural language processing (NLP), *entity labeling* aims to assign predefined labels to the semantic entities in VRDs (G. Jaume and Thiran, 2019; Liu et al., 2019a; Yu et al., 2020b), and *entity relation extraction*[3] predicts relations between these semantic entities. Compared with NER and relation extraction in the traditional NLP, KIE from VRDs is a more challenging task. First, a normal (named-) entity in plain text does not contain layout information as those semantic entities in VRDs. Second, normal relation extraction predicts the relation between two *given* mentions, while relation extraction in VRDs needs to predict the relation between *any* two semantic entities in the document. As Figure 1 (left) illustrates, the entity labeling task is to tag "533" with the label "Answer", "Registration No." with label "Question". However, which question can be answered by "533" remains unknown without *entity relation extraction*. Compared with labeling, the entity relation extraction task is less explored, but its benefits at least include: 1) pro-

---

[1]In different papers, the bounding boxes are called differently, such as semantic entities, text segments, etc. In this paper, we follow the naming convention used in the paper of the FUNSD dataset (G. Jaume and Thiran, 2019).

[2]Other visual features such as font size, colors and so on are not provided in the FUNSD dataset, thus not considered in this work.

[3]To avoid confusion with the entity linking in knowledge graphs, we name the task as entity relation extraction instead of entity linking used in G. Jaume and Thiran (2019), as this task aims to discover relations between semantic entities

---

*Corresponding author. The author's contributions were carried out while at Alibaba Group. His current affiliation is Vipshop (China) Co., Ltd.
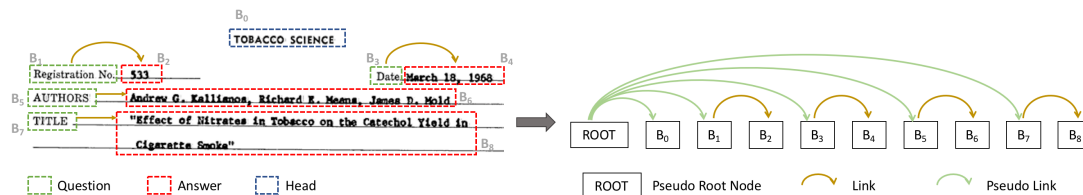
Figure 1: Left part shows one visually rich document from the FUNSD data. Group of words with one bounding box means one semantic entity and we number each entity as B_i. Different colors of boxes mean their different entity labels as legends under the example list. Relation links between semantic entities always point from key entities to value entities. We convert entity relations in the VRD to one tree and add the pseudo root node following the similar setting in dependency tree, and then link zero-head entities to the pseudo node as the pseudo links shown in the right part.

viding additional structural information closer to human comprehension of the VRDs, and 2) being easier to be transferred to other domains when the predefined label set changes. Therefore, in this paper, we concentrate on the task of semantic entity relation extraction which discovers the relation between two groups of words with layout information, as the yellow links in Figure 1 (left) show.

As another similar task to entity relation extraction in VRDs, dependency parsing aims to find out syntactic relations between words of an input sentence, which has been widely studied for decades. Both of these two tasks capture pairwise relationship between basic units of the input data. We adapt the popular biaffine dependency parser which utilizes the biaffine attention to compute scores between words (Dozat and Manning, 2017) into the entity relation extraction task due to their similarity. Since visual features play an important role in the VRDs, we introduce abundant layout information into different layers of the model to enhance the original text-only biaffine model:

- At the **entity representation** layer, we use the LayoutLM (Xu et al., 2020b) to encode both the word group and coordinates.

- At the **document encoder** layer, we utilize graph convolutional networks (GCN) to combine textual and visual information in VRDs by mapping layout information into graph edge representation between entities (Liu et al., 2019a; Yu et al., 2020b).

- At the **relation scorer** layer, we extract relative position features between entities according to their coordinates.

Apart from the above, inspired by the joint POS tagging and dependency parsing model (Nguyen

and Verspoor, 2018), we propose the multi-task learning for both entity labeling and relation extraction to further improve the performance.

Abundant detailed experiments are conducted to verify our approach of applying the biaffine dependency parser to semantic entity relation extraction task in VRDs. Our proposed relation extraction model achieves 65.96% F1 score on the FUNSD dataset, demonstrating the effectiveness of our model. As for the real-world application scenario, our model has also been applied to the in-house customs data, achieving reliable performance in the production setting.

The contributions of this paper are as follows:

- We adapt the biaffine model used in dependency parsing to the entity relation extraction task and achieve 65.96% F1 score on the FUNSD dataset.

- We conduct detailed experiments to compare different representations of the semantic entity, different VRD encoders, and different relation decoders to better understand this task.

- We apply our model to the real-world customs data with different layouts and achieve high performance in the production setting.

2 Related Work

Visually rich document understanding includes many tasks, such as layout recognition (Zhong et al., 2019b; Li et al., 2020), table detection and recognition (Li et al., 2019a; Zhong et al., 2019a) and key information extraction (Graliński et al., 2020; Guo et al., 2019; Huang et al., 2019; G. Jaume and Thiran, 2019; Majumder et al., 2020). Our paper focuses on the key information extraction task which contains two subtasks, entity labeling and relation extraction. The former subtask
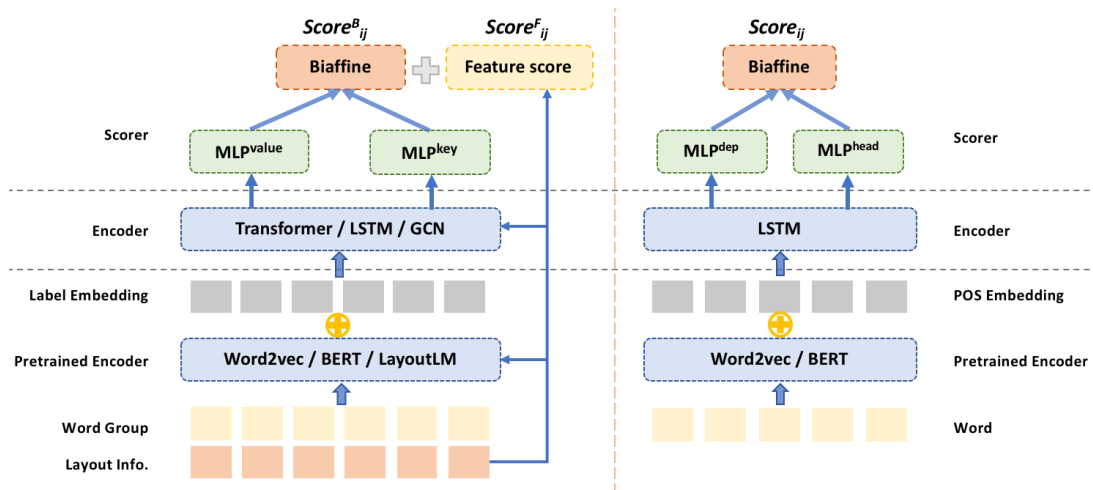
Figure 2: The architecture of our proposed entity relation extraction model (left) and the biaffine parser model (right).

tags entities with predefined labels, such as Task 3 on the SROIE data released by Huang et al. (2019), while the latter discovers relations between entities, such as Subtask C(3) on the FUNSD data (G. Jaume and Thiran, 2019).

To encode the semantic entity in VRDs, Yu et al. (2020b) and Wei et al. (2020) replace BiLSTM (Bi-directional Long Short-Term Memory) used by Liu et al. (2019a) with BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019b). Xu et al. (2020b) propose LayoutLM, which adds the 2-D position embedding into language model based on BERT and pretrain their language model on large-scale scanned document images with more visually-related loss function. Experiments verify that encoding the word group and layout coordinates at the same time is more effective for VRD understanding. LayoutLMv2 additionally introduces visual embedding into input layer and integrates a spatial-aware self-attention mechanism into the Transformer architecture (Xu et al., 2020a). And LayoutLMv2 performs better than LayoutLM in downstream VRD understanding tasks.

While encoding VRDs, previous works take entity labeling task as sequence labeling and re-implement the named entity recognition (NER) framework (Lample et al., 2016) but ignore layout information. Then, many works introduce a GCN-based module to encode layout information and combine textual and visual information together (Liu et al., 2019a; Yu et al., 2020b; Wei et al., 2020; Carbonell et al., 2021). In the GCN module, Liu et al. (2019a); Yu et al. (2020b) take layout features between entity b_i and b_j as edge embedding to up-

date entity representation while Wei et al. (2020) prune irrelevant nodes in graph according to same x-axis or y-axis coordinates to get the adjacency matrix.

To predict relations between entities, G. Jaume and Thiran (2019) provide one simple approach which concatenates the representations of two entities and use multi-layer perceptron (MLP) to obtain the relation score between entities. Carbonell et al. (2021) also use the MLP scorer but take GNN as document encoder instead of BERT and perform better. In the field of dependency parsing, Dozat and Manning (2017) propose the biaffine attention mechanism to compute scores between words, and achieve better performance than the MLP mechanism used by Kiperwasser and Goldberg (2016). As the biaffine attention is widely used in other tasks like NER (Yu et al., 2020a) and semantic role labeling (Li et al., 2019b), we propose to use it for the entity relation extraction task in this work.

3 Entity Relation Extraction as Dependency Parsing

Both semantic entity relation extraction and dependency parsing tasks aim to decide whether there exists relation between two entities/words and assume that links always point from key/head unit to value/modifier unit shown in Figure 1. Therefore, we can draw lessons from the dependency parsing exploration as it has been studied for several decades and achieved great progress. Biaffine parser, a strong model in dependency parsing, achieves competitive performance and is widely used in different scenes and tasks. This section

introduces how to apply the biaffine parser to our relation extraction task according to their similarity and difference.

3.1 Task Definition

Each scanned visually rich document is composed of a list of semantic entities, and each entity composes of a group of words and coordinates of the bounding box, defined as $b_i = \{[w_i^1, \ldots, w_i^m], [x_i^1, x_i^2, y_i^1, y_i^2]\}$, where $[w_i^1, \ldots, w_i^m]$ mean the word group, x_i^1/x_i^2 and y_i^1/y_i^2 are left/right x-coordinates and top/down y-coordinates respectively. Documents in our used dataset are annotated with the label of each entity and relations between entities. We represent each annotated document as $D = \{[b_1, \ldots b_n], [l_1, \ldots, l_n], [(b_1, b_{h_1}), \ldots, (b_m, b_{h_m})]\}$, where $l \in L$ is the label of each entity and L is the predefined entity label set. (b_i, b_{h_i}) mean the relation between entities b_i and b_{h_i}, and the link points from b_{h_i} to b_i. Notably, the entity may exist relations with more than one entity or does not have relation with any other entities.

3.2 Biaffine Parser

As Figure 2 (right) shows, biaffine dependency parser takes word and POS-tag embedding as the word representation, and uses multi-layer BiLSTM to encode the input sentence. Then, two MLP modules are used to strip away information not relevant to the current link decision. At last, the biaffine attention mechanism is utilized to compute the score of the dependency link between words.

We explore various aspects of applying the biaffine parser to our relation extraction in VRDs due to their similarity. Especially, we take the layout information into consideration besides the text itself, compared to the regular dependency parsing. In our proposed entity relation extraction model, we exploit important layout information at different processing levels, including entity encoder, document encoder and relation scorer, as Figure 2 (left) shows. We name our proposed model as SERA (**S**emantic **E**ntity **R**elation extraction **A**s dependency parsing). Details of our proposed SERA are discussed in the following subsections.

3.3 Entity Representation

At the input layer, in order to obtain better entity representations, we compare different ways to encode the information of semantic entity, containing the word group and the layout features. In this

work, we take the advantage of widely used pretrained models, including context-free word vector from word2vec (Mikolov et al., 2013), contextualized representations from BERT and LayoutLM. Specially, LayoutLM introduce coordinate information from bounding boxes during pretraining which is very suitable for our scenario.

In addition, we make use of the label of each semantic entity, such as "Question", "Answer" in FUNSD label set as Figure 1 shows. We map the entity labels into label embedding, as POS-tag embedding in dependency parsing. Then, we concatenate the entity representation and label embedding as the input of the document encoder for each semantic entity, as the following equation shows:

$$\mathbf{e}_i = \mathbf{b}_i \oplus \mathbf{l}_i \quad (1)$$

where $\mathbf{l}_i$ means entity label embedding, and $\mathbf{b}_i$ means the representation of semantic entity, which can be obtained from the above-mentioned three pretrained models, e.g. word2vec, BERT and LayoutLM .

3.4 Document Encoder

We compare different document encoders, including transformer, BiLSTM, and GCN, for better encoding the information of the semantic entities in VRDs. Specifically, we feed the representation of entity into the document encoder and obtain the output of the encoder as the contextual representation of the entity. Details of BiLSTM and transformer can refer to Lample et al. (2016) and Vaswani et al. (2017), respectively.

For GCN encoder, initial entity representation in graph is computed as subsection 3.3 shows. While updating the representation of the entity and edge, we follow the computation of Liu et al. (2019a). The edge embedding consists of 2 layout features, as the following equation shows:

$$\mathbf{r}_{i,j} = [x_{i,j}, y_{i,j}] \quad (2)$$

where x_{ij} and y_{ij} are horizontal and vertical distance between the two entity boxes respectively:

$$x_{i,j} = min(|x_i^1 - x_j^2|, |x_j^1 - x_i^2|)$$
$$y_{i,j} = min(|y_i^1 - y_j^2|, |y_j^1 - y_i^2|) \quad (3)$$

For entity b_i, we extract features $\mathbf{h}_{ij}$ of each neighbour b_j by concatenating the representation of the two entities and their corresponding edge.

$$\mathbf{h}_{ij} = \mathbf{e}_i \oplus \mathbf{r}_{i,j} \oplus \mathbf{e}_j \quad (4)$$

Then, we update the representation of entity and edge so that each entity can extract relevant information from other entities according to the document layout information, as the following equation shows:

$$\mathbf{e}_i^{'} = \sum_{j=1}^{n} \alpha_{ij}\mathbf{h}_{ij}$$

$$\mathbf{r}_{i,j}^{'} = \mathbf{W}^r\mathbf{h}_{ij} + \mathbf{b}^r$$

$$(5)$$

and α_{ij} is the attention weight and can be computed as follows:

$$\alpha_{ij} = \frac{exp(LeakyRelu(\mathbf{W}^a\mathbf{h}_{ij}))}{\sum_{j=1}^{n} exp(LeakyRelu(\mathbf{W}^a\mathbf{h}_{ij}))} \quad (6)$$

Where, n means the number of entities in a document.

3.5 Relation Scorer

Following the biaffine parser, we firstly apply MLP module to drop trivial information which is unrelated to current relation decision. Two MLP are used to generate the different representation of key and value roles in each relation link, which indicates the direction of arc in Figure 1.

$$\mathbf{h}_i^{key} = F(\mathbf{W}^{key}\mathbf{e}_i^{'} + \mathbf{b}^{key})$$
$$\mathbf{h}_j^{value} = F(\mathbf{W}^{value}\mathbf{e}_j^{'} + \mathbf{b}^{value})$$

$$(7)$$

where F is an activation function.

Then, we use biaffine attention to compute the score between two semantic entities as follows:

$$Score_{i,j}^B = \mathbf{h}_i^{key}\mathbf{W}_1^B\mathbf{h}_j^{value} + \mathbf{h}_i^{key}\mathbf{W}_2^B \quad (8)$$

Such biaffine mechanism can capture pairwise relationship between entities better.

We also use layout information $\mathbf{r}_{i,j}$ as external features to help the model predict relations between entities better. Such layout features indicate the position relationship between entity b_i and entity b_j: left-to-right or top-to-down. Empirically, we observe that entities in the left-to-right or top-to-down order are more likely to exist relations. We use MLP to compute the layout feature score as follows:

$$Score_{i,j}^F = \mathbf{W}^F\mathbf{r}_{i,j} + \mathbf{b}^F \quad (9)$$

Lastly, we add biaffine score with layout feature score together as the score of the relation between entity b_i and entity b_j:

$$Score_{i,j} = Score_{i,j}^B + Score_{i,j}^F \quad (10)$$

	Train			Test		
	#Docs	#Entities	#Rels	#Docs	#Entities	#Rels
FUNSD	149	7411	3902	50	2332	1048
W/ AUG	298	14,822	7804	50	2332	1048
CUSTOMS	1308	93,627	16,146	329	23,365	4305

Table 1: Data statistics of the FUNSD and customs datasets.

3.6 Relation Decoder

Based on relation scores between entities, two different relation decoding methods decide different loss functions of our training objective.

The first method is to judge whether there exists relation between any two entities in each VRD and such way is similar to semantic role labeling (SRL). In such setting, we take the relation prediction as a binary classification task and use binary cross entropy loss as G. Jaume and Thiran (2019).

The second is to choose one head entity from all entities in one VRD for the current one, which is similar to the decoder in dependency parsing. This method means that each entity must have exactly one head entity, namely single-head constraint. Now, relation prediction is seen as a multi-classification task and use softmax cross entropy loss as Dozat and Manning (2017).

4 Experiment Settings

4.1 Datasets

We conduct experiments on the FUNSD[4] data, which is published by G. Jaume and Thiran (2019) for the form understanding task. Moreover, to verify our proposed model, we also collect real-world dataset from the customs scenario.

FUNSD is composed of 199 fully annotated, scanned forms with comprehensive annotations to address form understanding tasks including entity labeling and relation extraction. We follow the data split as G. Jaume and Thiran (2019), and detailed data statistics are listed in Table 1, including the entity/relation distribution in FUNSD.

Customs Data consists of about 1,600 customs declaration documents in different layouts and languages collected by us. There are four types of documents: packing list, invoice, sales contract and customs declaration form, and each kind of document provides different information which is useful to apply to the customs. Figure 3 gives one invoice

---

[4]The FUNSD data can be downloaded from https://guillaumejaume.github.io/FUNSD/.

2763

Figure 3: One commercial invoice example in customs data and its filetype is Excel. Some sensitive information are replaced by blue blocks.

example, providing unit price, quantity and other details. Customs documents may be in Chinese or English, and their format may contain Word, Excel, PDF or image. We parse these documents by a self-developed OCR tool to get semantic entities in each VRD. We organize crowd-sourcing to annotate labels of entities given the predefined label set, containing 48 kinds of label which are important for customs information extraction system. We can get the key entities according to the map dictionary from each predefined entity label to its all possible names in VRDs due to these names are enumerable. Then, we link the entities from keys to values respectively with same labels. We finally get the annotated customs data annotated with labels of entities and relations between entities. The scale of our collected customs data is much bigger than the FUNSD data as Table 1 shows.

4.2 Data Preprocessing

Multi-Head & Zero-Head Entities. In terms of dependency parsing, one word must have one and only one head (the single-head constraint). However, zero-head entity that has no relations with any other entities or multi-head entity that has multiple heads does appear in our dataset. For zero-head entity, we add a pseudo root entity and link these zero-head entities to the pseudo entity as Figure 1 shows. For multi-head entity, we randomly remain one head entity and delete others to get single-head entity while under single-head constraint. In FUNSD, there are 324 (/4,236) and 16 (/1,048) multi-head entities in train/test data respectively, accounts for a small part in all data.

Auto Labels. It is intuitive that type-tagged entity will ease the prediction of relations between semantic entities. We employ an effective entity

labeling model consisting of two modules: entity encoder and label scorer. We take LayoutLM to encode the document and get the entity representation in a similar way as our relation extraction model. We also introduce three layout features w_i, h_i, c_i into our entity representation and map the features into 10-dim embedding. w_i and h_i mean the width and height of the bounding box and c_i means the length of characters in word group of each semantic entity. After concatenating the feature embedding and LayoutLM output, we pass them into the MLP scorer to compute the score of each candidate label. By this way, we get the auto label of our relation extraction data[5].

4.3 Evaluation Metrics

To evaluate our semantic entity linking model, we take the entity-level precision, recall and F1 score as measure standard. Under single-head constraint, we ignore the links pointed from pseudo root entity in gold and predicted results for fair comparison with other works.

4.4 Parameters

We investigate several pretrained language models to obtain entity representation, i.e, word2vec, BERT, and LayoutLM. For word2vec, we obtain the entity representation by averaging embeddings of the words contained in one entity; and for BERT/LayoutLM, we use the base model and take the hidden state output of the first subword of word group as the whole entity representation. Therefore, the representation dimension of words within bounding box is 100 while using word2vec and 768 while using BERT or LayoutLM. We use 100-dim embedding to represent the entity labels, so our entity representation is 200 or 864[6].

To encode the whole VRD, we investigate 1-layer BiLSTM or 1-layer transformer or 2-layer GCN encoders. The hidden state dimension of BiLSTM and transformer is 300 and the dimension of output edge and entity representations generated by GCN is 100.

The learning rate for BERT and LayoutLM is set to 1e-5 and others to 1e-2. The model are trained

---

[5]We train the labeling model on the whole training data and predict the auto labels of the test data. And we split training data into 5-fold, and train model with 4-fold to generate automatic labels of the left 1-fold training data.

[6]While using transformer, it's difficult to set the number of heads in multi-head self-attention if dim of entity representation is 868. Here, we use a 96-dim label embedding instead.

	P	R	F1
Previous works: our reimplementation			
BIAFFINE Dozat and Manning (2017)	0.0746	0.1131	0.0899
BERT G. Jaume and Thiran (2019)	0.2959	0.2872	0.2915
LAYOUTLM Xu et al. (2020b)	0.3041	0.3082	0.3062
GNN + MLP Carbonell et al. (2021)	-	-	0.39
Our models			
- FEATURE SCORER	0.6049	0.6020	0.6035
- AUTO LABEL	0.6046	0.6756	0.6381
SERA	0.6189	0.6756	0.6460
+ GOLD LABEL	0.7033	0.7576	0.7294
+ MTL	**0.6374**	0.6727	0.6546
+ MTL, AUG	0.6368	**0.6842**	**0.6596**

Table 2: Performance of entity relation extraction task on the FUNSD test data of previous works and our model with different but important settings. We re-implement previous works after application to entity relation extraction task, except for the work of Carbonell et al. (2021). We report their published experiment results in their paper.

	P	R	F1
WORD2VEC	0.0187	0.0403	0.0256
BERT	0.5539	0.5887	0.5708
LAYOUTLM	**0.6189**	**0.6756**	**0.6460**

Table 3: Performance of entity relation extraction on the FUNSD test data. We compare different pretrained language model used to encode entities and keep other modules of SERA unchanged.

for 50 iterations on FUNSD data and 100 iterations on customs data[7]. And each iteration we traverse the whole training data under all settings.

5 Experiment Results

5.1 Overall Results

We adapt the biaffine model from the dependency parsing task to our entity relation extraction task, and conduct detailed experiments on FUNSD dataset. Experimental results are shown in Table 2.

Previous works. Firstly, we train the original biaffine model (Dozat and Manning, 2017) after replacing each word and its POS-tag with word group and entity label of each entity, but achieve poor performance. Then, we re-implement the entity relation extraction model proposed by G. Jaume and Thiran (2019), which consists of BERT as the entity encoder and MLP as the relation scorer. And our re-implement results are much higher than the performance reported in their paper (0.04% F1). We replace BERT with LayoutLM and keep other parts unchanged to encode the layout coordinate information into relation extraction task and model performance improves a little. Carbonell et al. (2021) also utilize MLP link scorer but encode documents with k-layer GNN instead of BERT or LayoutLM,

and they achieve higher performance than other previous works.

Our Models. We propose our semantic entity relation extraction model based on the architecture of the biaffine parser as Section 3 describes. We apply LayoutLM/GCN as our entity/document encoder and optimize our models with softmax cross entropy loss under the single-head constraint. Results show that our proposed SERA achieves much higher performance than previous works by a large margin. Performance improvement demonstrates that layout information plays an important role in entity relation extraction task. Two ablation experiments verify the effectiveness of the layout feature scorer and auto labels of entities. Our SERA model can further improve the performance with two training strategies: data augmentation and multi-task learning of entity labeling and relation extraction.

Detailed experiments about our different explorations for entity relation extraction task are discussed in the following subsections.

5.2 Entity Representation

While encoding semantic entities, we employ three different pretrained models[8] and comparison experiment results are listed in Table 3. Results show that encoding entities with LayoutLM performs the best because it introduces layout information into its transformer encoder and has been pretrained on a large scale of VRDs compared with BERT. Word2vec achieves much poorer performance than the other two due to the missing context-aware information inside semantic entities.

Table 2 demonstrates that taking entity label embedding as part of the entity representation can improve the model performance. From our analysis on the FUNSD data, we find that many entity relation links point from entities with label 'Question' to entities with label 'Answer' and almost no

---

[7]We train the model for 50/100 iterations, and then predict test set on the trained model.

[8]In our model, BERT and LayoutLM encode entities in each document as we concatenate words in all bounding boxes in the order from top left to bottom right as Xu et al. (2020b).

	P	R	F1
TRANSFORMER	0.6022	0.6240	0.6129
BiLSTM	0.6110	0.6460	0.6280
GCN	**0.6189**	**0.6756**	**0.6460**

Table 4: Performance of entity relation extraction on the FUNSD test data. We compare different document encoder and other modules of SERA remains unchanged.

	P	R	F1
MLP + MULTI	0.5317	0.4803	0.5047
MLP + SINGLE	0.3041	0.3082	0.3062
BIAFFINE + MULTI	**0.6470**	0.3618	0.4641
BIAFFINE + SINGLE	0.6189	**0.6756**	**0.6460**

Table 5: Performance of SERA on the FUNSD test data. Different relation scorers with different loss functions are used under different constraints.

	P	R	F1
SERA	0.6189	0.6756	0.6460
+ MTL	**0.6374**	0.6727	0.6546
+ AUG	0.6315	**0.6851**	0.6572
+ MTL	0.6368	0.6842	**0.6596**
+ GOLD LABEL	0.7402	0.7777	0.7585

Table 6: Performance of entity relation extraction on the FUNSD test data. We train our SERA with two training strategies: data augmentation and multi-task learning.

performance gap as Table 5 shows. Using different relation scorer, the trend between these two decoders is contrary. MLP scorer performs poorer than biaffine scorer under single-head constraint. This is because biaffine scorer is more suitable for single-head constraint which has been proved by Dozat and Manning (2017). Without such constraint, the task is similar to SRL, and more SRL previous works prefer to the MLP scorer.

Our error analysis finds that biaffine scorer without single-head constraint leads to the model prefers to predict multi-head links for entities, which is not consistent with the data distribution.

Due to the big gap between the biaffine scorer with single-head constraint and the MLP scorer with multi-head constraint, we finally choose the biaffine scorer and add the single-head constraint in our experiments.

5.5 Training Strategies

To get better performance in entity relation extraction task, we apply two training strategies. Firstly, we take entity labeling and relation extraction tasks as multi-task learning (MTL) (Collobert and Weston, 2008; Nguyen and Verspoor, 2018) and these two tasks share the pretrained language model in entity encoder and fine-tune the sharing parameter together during training. Relation extraction task can improve by about 0.86% F1 while labeling model performance drops a little from Table 6. Performance improvement demonstrates that MTL is highly effective on alleviating error propagation from entity labeling task.

Secondly, we try to augment our training data due to the small size of training documents in the FUNSD data (Krizhevsky et al., 2017). We randomly drop some words in ratio 0.2 from word group of each entity to obtain more pseudo documents. We combine these pseudo training data with the gold training data and keep the test data unchanged. Models trained on the training data

relations in-between answers or questions[9]. Therefore, entity labels help the extraction model prune the unreasonable relations and enrich the entity representation. The gap between models with gold labels and auto labels is about 10% F1, which indicates the room for improvement is still large if we can obtain better auto entity labels.

5.3 Document Encoder

We investigate three popular encoders, including transformer, BiLSTM and GCN to encode VRDs. Experiment results in Table 4 show GCN encoder performs much better than the other two in our task. The GCN we applied can be seen as an improvement of self-attention mechanism due to it introduces layout information into the document encoder as Formula 4 shows. Such layout features indicate the positional relation between entities according to x-axis or y-axis coordinates. They help document encoder to copy more information from adjacent entities which are more relevant to current entity, while transformer updates entity representation according to the textual information alone. BiLSTM encodes the entities in documents in the plain sequential order. However, the sequential order is not suitable for the VRD understanding, for example, many key-value pairs in tables are in top and down order.

5.4 Relation Decoder

Decoding relation links between entities with or without the single-head constraint leads to a large

---

[9]In FUNSD training data, relations between answers or questions occupy only about 1%.

	P	**R**	**F1**
BERT			
ENGLISH	0.7726	0.7806	0.7853
CHINESE	0.7950	0.8320	0.8131
LayoutLM			
ENGLISH	**0.8464**	**0.8602**	**0.8533**

Table 7: Performance of entity relation extraction on the customs data using SERA. We compare different language models in different languages.

after augmentation improve performance by about 1.1% F1 with auto label and 2.9% F1 with gold label. Improvement gap between the two indicates that relation extraction model is sensitive to the accuracy of entity labels.

Combination of these two strategies performs the best under the auto entity label settings.

5.6 Customs Data

We apply our SERA with best configuration to our collected customs data. Due to documents in customs data may be in Chinese and English, pretrained Chinese or English language model cannot cover the words in documents by its vocabulary perfectly. We conduct experiments with different pretrained models in different languages to study this problem deeply.

As Table 7 shows, our proposed model works well on the customs data, whose scale is much larger than FUNSD. Customs data contain more layout information, such as tables as Figure 3 shows. We observe that the Chinese BERT is better than the English BERT on our language mixed data. We analyse their vocabularies and find Chinese vocabulary can cover more words in documents. Even though Chinese BERT performs better, English LayoutLM still achieves the best results among three pretrained models. This indicates encoding layout information into language model makes difference.

6 Conclusion

This paper focuses on the largely-unexplored entity relation extraction task in VRDs. We take advantages of previous works in semantic entity labeling and dependency parsing and propose our relation extraction model SERA. Our improved entity relation extraction model achieves 64.60% F1 score on the FUNSD data, outperforming previous baseline by a large margin; and we employ two simple but effective training strategies to further improve the

performance to 65.96%, i.e., multi-task learning with entity labeling and data augmentation. In addition, We verify the effectiveness of our model on the real-world customs data with different layouts in the production setting. In the future, we plan to incorporate more visual features into the relation extraction model and also extend it into more domains and business scenarios.

References

Manuel Carbonell, Pau Riba, Mauricio Villegas, Alicia Fornés, and Josep Lladós. 2021. Named entity recognition and relation extraction with graph neural networks in semi structured documents. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9622–9627. IEEE.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 160–167. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

H. Kemal Ekenel G. Jaume and J. Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (IC-DARW)*, volume 2, pages 1–6.

Filip Graliński, Tomasz Stanisławek, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2020. Kleister: A novel task for information extraction involving long documents with complex layout. *arXiv preprint arXiv:2003.02356*.

He Guo, Xiameng Qin, Jiaming Liu, Junyu Han, Jingtuo Liu, and Errui Ding. 2019. Eaten: Entity-aware attention for single shot visual text extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 254–259. IEEE.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. ICDAR2019 competition on scanned receipt OCR and information extraction. In *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*, pages 1516–1520.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Trans. Assoc. Comput. Linguistics*, 4:313–327.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.

Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2019a. Tablebank: Table benchmark for image-based table detection and recognition. *CoRR*, abs/1903.01949.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis. *arXiv preprint arXiv:2006.01038*.

Zuchao Li, Shexia He, Hai Zhao, Yiqing Zhang, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2019b. Dependency or span, end-to-end uniform semantic role labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6730–6737.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2019a. Graph convolution for multimodal information extraction from visually rich documents. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 32–39.

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

Bodhisattwa Prasad Majumder, Navneet Potti, Sandeep Tata, James Bradley Wendt, Qi Zhao, and Marc Najork. 2020. Representation learning for information extraction from form-like documents. In *proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 6495–6504.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 81–91.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Mengxi Wei, Yifan He, and Qiong Zhang. 2020. Robust layout-aware IE for visually rich documents with pre-trained language models. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2367–2376.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. 2020a. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020b. 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.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020a. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. 2020b. Pick: Processing key information extraction from documents using improved graph learning-convolutional networks. *arXiv preprint arXiv:2004.07464*.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2019a. Image-based table recognition: data, model, and evaluation. *arXiv preprint arXiv:1911.10683*.

Xu Zhong, Jianbin Tang, and Antonio Jimeno-Yepes. 2019b. Publaynet: largest dataset ever for document layout analysis. *CoRR*, abs/1908.07836.