Hypergraph based Understanding for Document Semantic Entity Recognition

Anonymous ACL submission

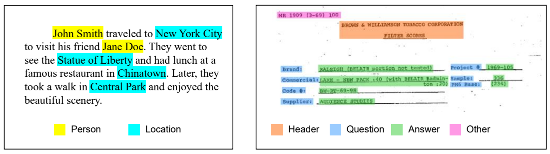
Abstract

Semantic entity recognition is an important task in the field of visually-rich document understanding. It distinguishes the semantic types of text by analyzing the position relationship between text nodes and the relation between text content. The existing document understanding models mainly focus on entity categories while ignoring the extraction of entity boundaries. We build a novel hypergraph attention document semantic entity recognition framework, HGA, which uses hypergraph attention to focus on entity boundaries and entity categories at the same time. It can conduct a more detailed analysis of the document text representation analyzed by the upstream model and achieves a better performance of semantic information. We apply this method on the basis of GraphLayoutLM to construct a new semantic entity recognition model HGALayoutLM. Our experiment results on FUNSD, CORD, XFUND and SROIE show that our method can effectively improve the performance of semantic entity recognition tasks based on the original model. The results of HGALayoutLM on FUNSD and XFUND reach the new state-of-the-art results.

1 Introduction

With the development of information technology, documents have become a main information carrier nowadays ,which contains kinds of information type, such as text, table and image. Manual recognition of these documents often requires plenty of manpower. OCR tools can only help us to identify the text, layout and other simple information in the document. To further understand documents, Visually-rich Document Understanding (VRDU) (Xu et al., 2020b) is proposed to make use of visual, textual and other information for more in-depth analysis.

Semantic Entity Recognition (SER) is an important task in the field of VRDU. Its purpose is to extract and classify the text with special semantic information in documents. Different from text sequences in traditional natural language processing tasks, the information in documents is not one-dimensional, single-modal and continuous, but two-dimensional, multimodal and discrete. It is necessary to analyze not only text information, but also other modal information such as layout and vision in the document. Figure 1 shows the difference between the traditional named entity recognition (NER) task on a single modal text and the semantic entity recognition task on a document. Firstly, the text form of a single modal text task is a fixed text sequence, while the discrete text in a document is composed of text nodes in different locations. Secondly, the named entity recognition task of a single modal text only needs to consider the semantic relationship between the tokens in the text sequence. However, the semantic entity recognition task on the document needs to consider not only the semantic relationship between nodes, but also the position relationship between nodes. Finally, the span range of entity tags of NER task is flexible, while the range of task tags of semantic entity recognition task on document is affected by nodes. Texts of the same node in the document share the same label in most cases.



(a) Named Entity Recognition Task
(b) Document Semantic Entity Recognition Task

Figure 1: Difference in Document Task.

With the development of pre-training technology, document pre-training model has become popular. LayoutLM (Xu et al., 2020b) is the first multi-modal pre-trained model to associate text with layout and vision, achieving leading results on multiple downstream document understanding tasks including semantic entity recognition. Subsequently, more multi-mode pretraining models, such as LayoutLMv2 (Xu et al., 2020a), BROS (Hong et al., 2022), ERNIE-Layout (Peng et al., 2022) and LayoutLMv3 (Huang et al., 2022) have been proposed successively. By integrating text, layout and visual information, they realize the understanding and information extraction of documents. So far, GraphLayoutLM (Li et al., 2023) and GeoLayoutLM (Luo et al., 2023) have the best performance in semantic entity recognition tasks. GraphLayoutLM achieves the best F1 score of 94.39 and 93.56 on the FUNSD (Jaume et al., 2019) and XFUND (Xu et al., 2021) datasets, and GeoLayoutLM achieves the best F1 score of 97.97 on the CORD (Park et al., 2019) datasets. However, these existing methods focus on the upstream document understanding part and pay little attention to the downstream task. GeoLayoutLM has studied the novel relational extraction head and achieves great improvement in the relational extraction task. But it has not done more research on the semantic entity recognition task. We study the problem of ignoring the downstream header and classification method in the semantic entity recognition task in the existing document intelligence work and propose a novel improvement scheme.

Traditional Semantic Entity Recognition. The traditional document semantic entity recognition task process is shown in (a) of the Figure 2. In document understanding process, text nodes are spliced into text sequences and become text token sequences of documents after tokenization. These text nodes will be transformed to the high-dimensional feature representations after the analysis of the document understanding model. To extract semantic information from document token features, linear layer or multilayer perceptron (MLP) will be used to convert high-dimensional features into label probabilities, and the training objective is cross entropy loss. Although this method can distinguish the node categories in the document, it ignores the characteristics of the document structure, and it is difficult to make the classification layer pay attention to the node span.

Hypergraph Semantic Entity Recognition. Inspired by Global Pointer (Su et al., 2022), we use the idea of hypergraph to extract the semantic information of documents and propose a Hypergraph Attention(HGA) strategy for document semantic entity recognition. (b) of the Figure 2 shows us the process of hypergraph semantic recognition. Different from the traditional classification method, the semantic entity recognition idea of HGA regard the document token features as graph nodes. The target entity is the set of nodes with the same hyperedge and the hyperedge type represents the entity label type. The process of hypergraph extraction is to establish hyperedges between token feature nodes. Besides, we use the span hyperedge encoding to add the span information of text nodes. Through the hypergraph and span position, header can better focus on the entity boundary information and establish the relationship between the document discrete text span and the entity boundary.

Our main contributions are as follows:

- We construct a novel hypergraph attention document semantic entity recognition method, HGA. It transforms the traditional token sequence classification problem into a hypergraph construction process. By establishing different types of hyperedges between text nodes, the header can extract semantic entities.
- We propose a novel span hyperedge position encoding and balanced hyperedge loss. Span hyperedge position encoding makes the header focus more on the same text span prompt during hyperedge construction. Balanced hyperedge loss can help to solve the problem of matrix sparsity caused by too many hyperedge types in some scenarios.
- We construct a novel document semantic entity recognition model HGALayoutLM based on the HGA method. The experiment results show that the model has good performance in the scene with few types of labels. HGALayoutLM has obtained the best results on the FUNSD, SROIE and XFUND datasets.

2 Related Work

In recent years, self-supervised pre-training technology has become the mainstream trend in the fields of natural language processing (NLP) and computer vision (CV). BERT (Devlin et al., 2018) is a classic pre-training model that has shown great effectiveness in various tasks such as question answering, natural language generation and text classification. Masked Language Modeling (MLM) is a significant pre-training task proposed by BERT

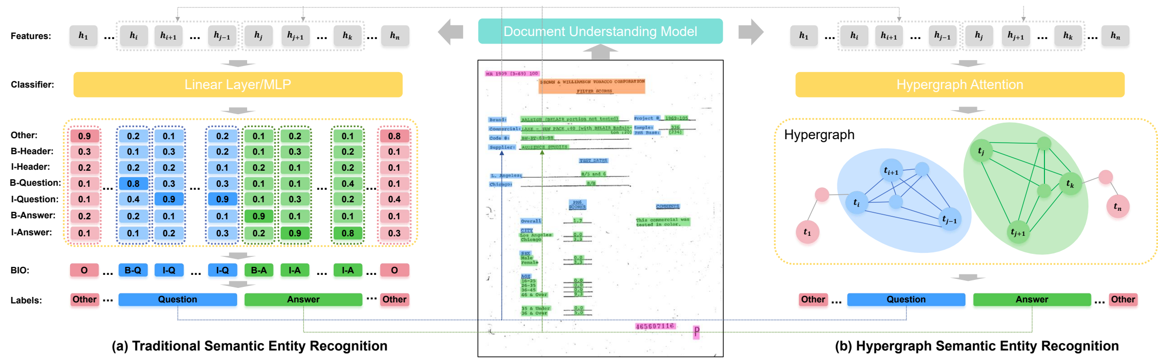


Figure 2: Traditional Semantic Entity Recognition and Hypergraph Semantic Entity Recognition.The document is from FUNSD dataset. Only the text sequence is shown in the figure. The rectangles with different colors in the figure are text nodes. The colors on the document nodes represent the different class labels. The orange color represents the label "HEADER". Blue is the label "QUESTION". Green is the label "ANSWER". Pink is the nonmeaning label, which is "OTHER".

that enables models to learn textual representations by predicting the raw vocabulary ids of randomly masked word markers based on context. Since then, a series of mask language models such as RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019) and XLNet (Yang et al., 2019) have been proposed successively. These models achieve good results on natural language understanding tasks.

However, the single modal language model can not understand documents with complex formats and diverse types well. To fully understand the content of complex documents, LayoutLM (Xu et al., 2020b) adds layout and document information on the basis of BERT to supplement the document format missing from plain text. Following LayoutLM, BROS (Hong et al., 2022), LayoutLMv2 (Xu et al., 2020a), XYLayoutLM (Gu et al., 2022), ERNIE-Layout (Peng et al., 2022), LayoutLMv3 (Huang et al., 2022) and other multimodal pre-training document understanding models have been proposed successively and constantly make breakthroughs in various tasks in the field of document understanding. These models understand the document through the fusion of text, layout and vision information. Since document nodes are suitable to be represented by graph structures, some works begin to apply graph structures to document understanding models, such as ERNIE-mmLayout (Wang et al., 2022), ROPE (Lee et al., 2021), FormNet (Lee et al., 2022), and GraphLayoutLM (Li et al., 2023).

The latest GraphLayoutLM and GeoLayoutLM (Luo et al., 2023) are both built on the basis of LayoutLMv3. They have achieved the most excellent results in several tasks of document information extraction. GraphLayoutLM models the document structure based on the hierarchical and positional layout of the document and represents the document layout modeling with a graph structure. To integrate graph structure information into the process of document understanding, GraphLayoutLM proposes graph reordering and graph masking strategies, adding graph information into the document understanding model in the form of sequence and self-attention mask. GeoLayoutLM implements geometric pre-training to enrich and enhance feature representation through three specially designed geometry-related pre-training tasks. In addition, GeoLayoutLM uses a novel relation header in the fine-tuning phase and obtains a big improvement over LayoutLMv3 in the relation extraction task. At present, little attention is paid to the effects of downstream task heads on the performance of various types of tasks. GeoLayoutLM proposes a novel relational header, but there is still a lack of research on the downstream task of semantic entity recognition in the field of document understanding. Most of the current models use a linear layer and cross-entropy to predict BIO label probabilities when dealing with semantic entity recognition tasks, such as LayoutLM, BROS, LayoutLMv2, etc. LayoutLMv3 and its derived models utilize a linear layer in the few label case and employ MLP when number of label types is large. These approaches are fundamentally the same. Differently, UDop (Tang et al., 2023) is a new unified document intelligent framework, which adopts encoder-decoder structure. However, the decoder will cost a large computational cost. Taking inspiration from Global Pointer (Su et al., 2022), we design a simple hypergraph header that incorporates document span information to achieve better SER task performance.

3 Methodology

3.1 Overview

The process of semantic entity recognition based on Hypergraph Attention is shown in Figure 3. Different from traditional semantic entity recognition methods, HGA focuses on extracting special entities. Instead of using BIO labels as annotations for model input, we use each special labels. Labels without semantics are no longer considered as an entity label type. HGA regards token features as unit nodes, and the process of establishing hyperedges between tokens can realize the extraction of special entities. It is worth noting that the node referred to here correspond to each token of token sequence. Text nodes, as mentioned earlier, are discrete pieces of text at different locations in the document. A text node corresponds to one or more token feature nodes. The process of hyperedge extraction can realize the extraction of special semantic entites and classification of different entity labels. An entity without any hyperedge connection is an entity with no special semantic, which is regarded as an Other label in BIO labeling.

To assist the construction of hyperedges, we use the span of each text node to generate the span position corresponding to the feature sequence. Then we use the span position encoding to add span information to the hypergraph construction process. In this way, the model can divide the hyperedge according to the text node span, so as to achieve more accurate extraction of the special entity range. In the stage of semantic entity extraction, we use multi-label classification to determine whether a node is connected by a hyperedge. Since there may be more than one type of hyperedges satisfying the join condition. To ensure the uniqueness of the entity type, we select the hyperedge with the maximum probability to establish the connection based on multi-label classification result.

3.2 Hypergraph Attention Header

We use the multi-head self-attention to represent the hypergraph. Consider a hypergraph with L number of nodes and N class of hyperedges. We use a multihead attention score of shape $N \times L \times L$ as the representation of this hypergraph. Hyperedge classes are represented by different heads of multi-head attention. The attention matrix corresponding to each head represents the distribution of a type hyperedge.

In the hypergraph, each token corresponds to a node. Assume the document token sequence is $x = \{x_1, x_2, ...x_n\}$. After understanding the document model, we convert the input token sequence into a high-dimensional feature representation sequence of the tokens:

$$h = \{h_1, h_2, ...h_n\} = DocModel(\{x_1, x_2, ...x_n\}), \tag{1}$$

where $h \in \mathbb{R}^{L \times H}$ is the high-dimensional feature representation sequence of the token and $DocModel$ is the document understanding model. L indicates the token sequence length, which also represents the number of token nodes. H is the feature dimension size. Based on h, we can obtain the query vector q and the key vector k:

$$\begin{aligned} q = \{q_\alpha : W_{q,\alpha} h + b_{q,\alpha}\}, \\ k = \{k_\alpha : W_{k,\alpha} h + b_{k,\alpha}\}, \end{aligned} \tag{2}$$

where $\alpha \in \mathbb{Z}^D$ is one head in multi-head attention, which can be regarded as a type in D kinds of hyperedges. With multi-head query vector and key vector, hypergraphs can be represented by a self-attention score calculated by q and k:

$$s = q^T k = \{s_\alpha(i,j) : q_{i,\alpha}^T k_{j,\alpha}, i \in \mathbb{Z}^L, j \in \mathbb{Z}^L\}. \tag{3}$$

$s_\alpha(i,j)$ is the attention score at the α type hyperedge span with $[i,j]$. $q_{i,\alpha}$ and $k_{j,\alpha}$ are the start and end of the span with $[i,j]$ in the α type hyperedge matrix. In this way, we implement hypergraph extraction of semantic entities.

3.3 Span Position Encoding

As we mentioned in Introduction, tokens of the same text node normally share the same semantic label in the process of semantic entity recognition of documents. We hope that the header can consider this span boundary prompt during entity extraction. Therefore, we construct the span position of the token sequence based on the text nodes and incorporate span information into the headers through position encoding. As shown in Figure 3, token feature sequence $h\{h_1, h_2, ...h_n\}$ and text node sequence $N = \{N_0, N_1, ...N_m\}$ has a surjective relation. We define this relational mapping as:

$$f(h_i) = N_j, h_i \in h, N_j \in N. \tag{4}$$

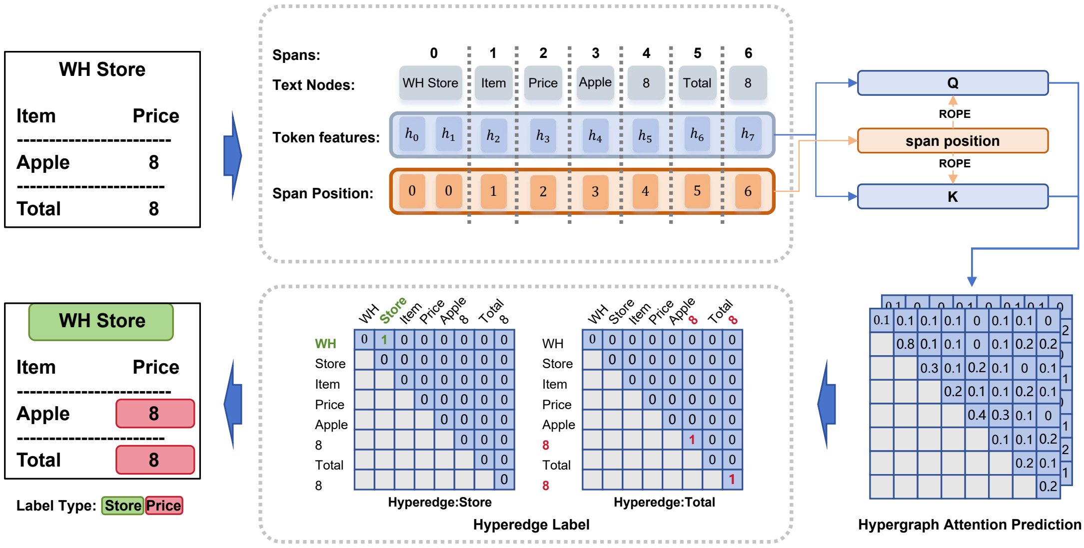


Figure 3: Semantic Entity Recognition Process Based on Hypergraph Attention. Only the text processing part of the model is shown in the figure. In the span position generation stage, the span position of the token feature sequence needs to be created by using the text node range span. The token features will be linearly transformed and encode the span position into a query vector Q and a key vector V. The multi-head hypergraph attention score is calculated from Q, V and added with the lower triangle mask. We regard each attention head as a sub-hypergraph corresponding to each hyperedge type.

Based on this relation mapping, we construct the span position. For the same text node N_j, All token feature nodes that have a mapping relationship with the same text node N_j share the same position:

$$\begin{aligned} p_i &= Position(f(h_i)) \\ &= Position(N_j) \\ &= j, h_i \in h, N_j \in N, \end{aligned} \tag{5}$$

where p_i is the span position of token feature h_i, $Position$ is the index of N_j. In this way, we can obtain the span position sequence $p = \{p_1, p_2, ...p_n\}$. On the basis of p, we use rotary position coding (Su et al., 2021) to generate position encoding $\mathcal{R}$, which satisfies $\mathcal{R}_i^T \mathcal{R}_j = \mathcal{R}_{j-i}$. Then the calulation of multi-head hypergraph score will be adjust to the following form:

$$\begin{aligned} s_\alpha(i,j) &= (\mathcal{R}_i q_{i,\alpha})^T (\mathcal{R}_j k_{j,\alpha}) \\ &= q_{i,\alpha}^T \mathcal{R}_i^T \mathcal{R}_j k_{j,\alpha} \\ &= q_{i,\alpha}^T \mathcal{R}_{j-i} k_{j,\alpha}. \end{aligned} \tag{6}$$

Because the start is always before the end when the span of token sequence is extracted. Span extraction nodes should not appear in the lower triangular region of the hypergraph attention score. For the purpose of making the hyperedge construction more reasonable, we add m_{tril} to the hypergraph matrix and the final hypergraph score format is as follow:

$$s_\alpha(i,j) = q_{i,\alpha}^T \mathcal{R}_{j-i} k_{j,\alpha} + m_{tril}(i,j). \tag{7}$$

3.4 Balanced Hyperedge Loss

In the process of loss calculation, we collect positive samples P_α and negative samples N_α respectively for each type of hyperedge α . The positive sample indicates that there is a α type hyperedge span with $[i,j]$ in α type hypergraph, while the reverse is a negative sample. The formats of P_α and N_α are as follows:

$$\begin{aligned} P_\alpha &= \{s_\alpha(i,j) | l_\alpha(i,j) = 1\}, \\ N_\alpha &= \{s_\alpha(i,j) | l_\alpha(i,j) = 0\}, \end{aligned} \tag{8}$$

where l is the hypergraph label matrix corresponding to s. With the sets of positive and negative samples,we can get the positive sample loss $\mathcal{L}_p$ and the negative sample loss $\mathcal{L}_n$:

$$\begin{aligned} \mathcal{L}_p &= \log\left(1 + \sum_{(i,j)\in P_\alpha} e^{-s_\alpha(i,j)}\right), \\ \mathcal{L}_n &= \log\left(1 + \sum_{(i,j)\in N_\alpha} e^{s_\alpha(i,j)}\right). \end{aligned} \tag{9}$$

Different from Global Pointer (Su et al., 2022), we gain the final loss with a balance factor $b \in [0,1)$

to avoid the matrix sparsity caused by too many label types. The final training loss of hypergraph attention score can be expressed in the following form:

$$\mathcal{L} = (1+b)\mathcal{L}_p + (1-b)\mathcal{L}_n. \quad (10)$$

3.5 HGALayoutLM

To verify the performance of the HGA method, we applied HGA to the latest GraphLayoutLM to build a novel semantic entity recognition model, HGALayoutLM. Consistent with GraphLayoutLM, we leverage the hierarchical layout of documents to build a hierarchical tree. Then we add position relationships between sibling nodes in the tree to construct the document structure graph G. The text nodes will be sorted according to the hierarchical and position relationship of G before concatenation to obtain a more reasonable reading order. In addition, we follow the architecture of GraphLayoutLM and add a graph mask layer to model to encode the relation information in G into the self-attention score.

Based on the graph structure-prompted document understanding model, we use the hypergraph attention layer as the header for document semantic entity recognition. The feature sequence of the token and the generated span position are used as the header input. The HGA method is used to help the model extract and classify semantic entities according to the text node span prompts.

4 Experiment

4.1 Experimental Setup

Model Settings. The model settings are consistent with those of GraphLayoutLM. The text sequence length is 512 and the document image is resized to $3 \times 224 \times 224$ dimensions. The image is cut into 196 patches in the size of 16×16. Transformer self-attention layer scaling factor α is set to 32. For HGALayoutLM$_{\text{BASE}}$, the hidden layer dimensions, the number of encoder self-attention layers, the number of self-attention heads and intermediate dimensions for feed-forward networks are set to 768,12,12 and 3072, respectively. The head number of graph mask layer is 6. The hidden layer dimension, encoder self-attention layer number, self-attention head number and intermediate dimensions for feed-forward networks of HGALayoutLM$_{\text{LARGE}}$ are set to 1024,24,16 and 4096, respectively. The head number of graph mask layer is 8. The hidden size of hypergraph attention layer in both base and large model is set to 64. To ensure the fairness of the experiment, we convert the results of hypergraph extraction into the format of BIO annotations for comparison.

Datasets. We select four commonly used document information extraction datasets. Three of these datasets are in English, including FUNSD, CORD and SROIE. One is the Chinese dataset, XFUND. The current XFUND task semantic entity recognition task of comparative experiment results is less, and there is almost no LARGE version experiment results. We only choose the BASE version of the model for our experiments. Detailed dataset information and hyper-parameters settings can be viewed in Appendix A.1 and Appendix A.2.

Baselines. We choose the classical natural language processing model BERT (Devlin et al., 2018) as the single modal document understanding comparison model and select several classical multimodal document understanding models, such as LayoutLM (Xu et al., 2020b), BROS (Hong et al., 2022), LayoutLMv2 (Xu et al., 2020a) and LayoutXLM (Xu et al., 2021). We also include the latest works in document understanding for comparison, such as ERNIE-Layout (Peng et al., 2022), LayoutLMv3 (Huang et al., 2022), GeoLayoutLM (Luo et al., 2023), GraphLayoutLM (Li et al., 2023) and UDop (Tang et al., 2023).

4.2 Main Results

The English datasets experiment results are shown in Table 1. The BASE version of HGALayoutLM using hypergraph attention layer as the header has achieved the best results on FUNSD and SROIE datasets (94.32 on FUNSD and 99.53 on SROIE), even when compared to the LARGE versions of models. Compared with GraphLayoutLM$_{\text{BASE}}$ using linear classification, HGALayoutLM achieves improvements of 0.89, 0.39 and 0.54 on FUNSD, CORD and SROIE datasets, respectively. The LARGE version of HGALayoutLM has achieved F1 scores of 95.31 and 99.61 on FUNSD and SROIE respectively, further updating the best performance on these datasets. Compared with GraphLayoutLM in the LARGE version, HGALayoutLM has F1 score 1.15 and 0.19 higher on FUNSD and SROIE datasets, respectively. This demonstrates the effectiveness of HGA on the task of less labels.

Table 1: Precision, Recall and F1 Score of Results on FUNSD, CORD, SROIE Datasets. Model labeled with "†" indicate that its results are obtained through replication in our experiments. Since some predictions on the web based on LayoutLMv3 on the SROIE dataset are completely correct, we do not list the results on SROIE as the state of the art.

Model	Header	FUNSD			CORD			SROIE		
		P	R	F	P	R	F	P	R	F
$\text{BERT}_{\text{BASE}}$	Linear	54.69	67.10	60.26	88.33	91.07	89.68	90.99	90.99	90.99
$\text{LayoutLM}_{\text{BASE}}$	Linear	75.97	81.55	78.66	94.37	95.08	94.72	94.38	94.38	94.38
$\text{BROS}_{\text{BASE}}$	Linear	81.16	85.01	83.05	-	-	96.50	-	-	96.28
$\text{LayoutLMv2}_{\text{BASE}}$	Linear	80.29	85.39	82.76	94.53	95.39	94.95	96.25	96.25	96.25-
$\text{LayoutXLM}_{\text{BASE}}$	Linear	-	-	79.40	-	-	-	-	-	-
XYLayoutLM	Linear	-	-	83.35	-	-	-	-	-	-
$\text{LayoutLMv3}_{\text{BASE}}$	Linear/MLP	90.82	91.55	91.19	96.35	96.71	96.53	-	-	99.25
$\text{GraphLayoutLM}_{\text{BASE}}$	Linear/MLP	92.46	**93.85**	93.15	97.02	**97.53**	97.28	-	-	99.30
$\text{GraphLayoutLM}^{\dagger}_{\text{BASE}}$	Linear/MLP	93.62	93.25	93.43	96.87	97.38	97.13	98.40	**99.58**	98.99
$\text{HGALayoutLM}_{\text{BASE}}$	HGA	**94.84**	93.80	**94.32**	**97.89**	97.16	**97.52**	**99.58**	99.48	**99.53**
$\text{BERT}_{\text{LARGE}}$	Linear	61.13	70.85	65.63	88.86	91.68	90.25	92.00	92.00	92.00
$\text{LayoutLM}_{\text{LARGE}}$	Linear	75.69	82.19	78.95	94.32	95.54	94.93	95.24	95.24	95.24
$\text{BROS}_{\text{LARGE}}$	Linear	82.81	86.31	84.52	-	-	97.28	-	-	96.62
$\text{LayoutLMv2}_{\text{LARGE}}$	Linear	83.24	85.19	84.20	95.65	96.37	96.01	99.04	96.61	97.81
$\text{ERNIE-Layout}_{\text{LARGE}}$	Linear	-	-	93.12	-	-	97.21	-	-	97.55
$\text{LayoutLMv3}_{\text{LARGE}}$	Linear/MLP	91.51	92.70	92.10	97.45	97.52	97.49	-	-	-
UDop	Decoder	-	-	92.08	-	-	97.58	-	-	-
GeoLayoutLM	Linear/MLP	-	-	92.86	-	-	**97.97**	-	-	-
$\text{GraphLayoutLM}_{\text{LARGE}}$	Linear/MLP	94.49	94.30	94.39	97.75	**97.75**	97.75	-	-	-
$\text{GraphLayoutLM}^{\dagger}_{\text{LARGE}}$	Linear/MLP	94.37	93.95	94.16	97.32	97.68	97.50	99.27	**99.58**	99.42
$\text{HGALayoutLM}_{\text{LARGE}}$	HGA	**95.67**	**94.95**	**95.31**	**97.97**	97.38	97.67	**99.69**	99.53	**99.61**

However, we can find that the performance of HGA is not outstanding on the CORD dataset. We think this is because the CORD dataset has a large number of label categories. The number of labels in CORD is an amazing 30, compared with the 3 or 4 label categories in other datasets. Since in the process of constructing the hypergraph, different types of hyperedges are built separately. Plenty of label categories will make the effective span nodes of hypergraph matrix sparse, which is not conducive to semantic entity recognition. However, by comparing GraphLayoutLM, we can find that HGA header can still improve the performance.

The experiment results of XFUND dataset are shown in Table 2. We can find that our HGALayoutLM has achieved the state of the art in XFUND. This further verifies the effectiveness of HGA header.

4.3 Ablation Study

To verify the effectiveness of our Span Position Encoding. We conduct ablation study on FUNSD. We can see from Figure 4 that the entity extraction effect without position encoding(w/o pos) is much worse than that with position encoding. In addition, we also compare the performance of our span position encoding(w/ span pos) with that of traditional position encoding(w/ pos). We can find that the performance of our span position encoding is obviously better than that of traditional position encoding.This demonstrates the effectiveness of our span position encoding with span prompt.

In order to prove that Balanced Hyperedge Loss can solve the problem of sparse hyperedge matrix caused by too many entity types. We conduct exper-

Table 2: Precision, Recall and F1 Score of Results on XFUND Datasets. Model labeled with "†" indicate that its results are obtained through replication in our experiments.

Model	Header	XFUND		
		P	R	F
$\text{LayoutXLM}_{\text{BASE}}$	Linear	-	-	89.24
XYLayoutLM	Linear	-	-	91.76
$\text{LayoutLMv3}_{\text{BASE}}$	Linear	89.80	94.35	92.02
$\text{GraphLayoutLM}_{\text{BASE}}$	Linear	91.80	95.38	93.56
$\text{GraphLayoutLM}^{\dagger}_{\text{BASE}}$	Linear	92.30	94.69	93.48
$\text{HGALayoutLM}_{\text{BASE}}$	HGA	**92.79**	**95.70**	**94.22**

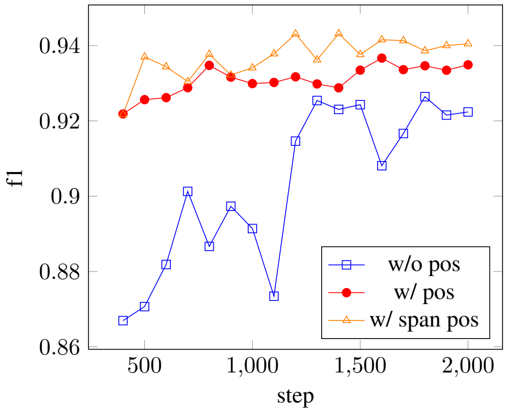


Figure 4: Position Encoding Comparison Line Chart. In order to highlight the contrast effect, we omit the results for the first 300 steps when the model has not converged.

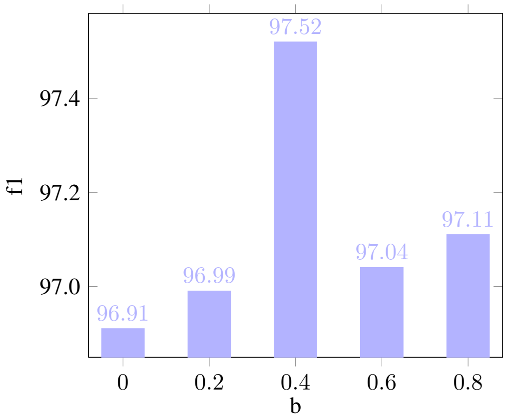


Figure 5: Further Study of Balance Factor.

iment statistics on different value of balance factor on CORD dataset with plenty of entity types and present the results in Figure 5. We can see that the performance of the unbalanced model ($b = 0$) is not ideal, even worse than the performance of the MLP header. However, proper balance factor allow the model to pay more attention to the hyperedge entities and achieve better results. For example, the performance when b is 0.4 exceeds the performance when the MLP layer is used as the header.

4.4 Anaysis of Different Header

To analyze the effects of different header, we adopt GraphLayoutLM$_{\text{BASE}}$ and HGALayoutLM$_{\text{BASE}}$ as the base model to conduct comparative experiments on three different headers, linear layer, MLP and HGA. The experiments are carried out on FUNSD, CORD, SROIE and XFUND datasets.

The experiment results are shown in Table 3. As the simplest network structure, the linear layer has the worst classification effect. The MLP layer increases the number of linear layers on top of the linear layer. It also joins activation layers and dropout layers to linear layers. The more complex network structure makes MLP slightly better than the semantic entity recognition of a single linear layer on most datasets. As our proposed hypergraph attention method, HGA performs significantly better than the other two classifiers,which shows the effectiveness of HGA, which demonstrates the superior performance of HGA.

Table 3: F1 Score of Different Header.

Header	FUNSD	CORD	SROIE	XFUND
Linear	93.48	96.98	98.99	93.03
MLP	93.58	97.13	99.28	93.48
HGA	94.32	97.52	99.53	94.22

To test the complexity of HGA, we compare HGALayoutLM with the model with traditional headers. The number of entity types is set to 3. As we can see from Table 4, HGA does not bring a large cost of time and space calculation and HGA is even less costly than MLP layer in terms of time and space computation.

Table 4: Analysis of Time and Space Complexity.

Model	Header	Params	Flops
GraphLayoutLM	Linear	88.02M	63.03G
GraphLayoutLM	MLP	88.61M	63.45G
HGALayoutLM	HGA	88.31M	63.24G

5 Conclusion

In this work, we propose a semantic entity recognition method (HGA) based on hypergraph attention. This method extracts semantic information from documents by establishing different hyperedges between feature nodes. On the basis of the hypergraph, we design span position encoding and balanced hyperedge loss to enhance the entity extraction capability of the hypergraph attention header. We use the HGA method to build a novel semantic entity recognition model HGALayoutLM based on GraphLayoutLM. This model has good performance in SER tasks. Experiments show that our method achieves the state of art on semantic entity recognition tasks on the FUNSD and XFUND datasets.

6 Limitation

When there are more types of semantic entities, the cost of improvement from HGA becomes higher. The number of superedge matrices increases because of more semantic entity categories. This not only leads to sparse matrix labels, but also to more model parameters.

References

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*.

Zhangxuan Gu, Changhua Meng, Ke Wang, Jun Lan, Weiqiang Wang, Ming Gu, and Liqing Zhang. 2022. Xylayoutlm: Towards layout-aware multimodal networks for visually-rich document understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4583–4592.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2022. Bros: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10767–10775.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091.

Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. Formnet: Structural encoding beyond sequential modeling in form document information extraction. *arXiv preprint arXiv:2203.08411*.

Chen-Yu Lee, Chun-Liang Li, Chu Wang, Renshen Wang, Yasuhisa Fujii, Siyang Qin, Ashok Popat, and Tomas Pfister. 2021. Rope: reading order equivariant positional encoding for graph-based document information extraction. *arXiv preprint arXiv:2106.10786*.

Qiwei Li, Zuchao Li, Xiantao Cai, Bo Du, and Hai Zhao. 2023. Enhancing visually-rich document understanding via layout structure modeling. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 4513–4523.

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*.

Chuwei Luo, Changxu Cheng, Qi Zheng, and Cong Yao. 2023. Geolayoutlm: Geometric pre-training for visual information extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7092–7101.

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. In *Workshop on Document Intelligence at NeurIPS 2019*.

Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, et al. 2022. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. *arXiv preprint arXiv:2210.06155*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. 2022. Global pointer: Novel efficient span-based approach for named entity recognition. *arXiv preprint arXiv:2208.03054*.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264.

Wenjin Wang, Zhengjie Huang, Bin Luo, Qianglong Chen, Qiming Peng, Yinxu Pan, Weichong Yin, Shikun Feng, Yu Sun, Dianhai Yu, et al. 2022. Ernie-mmlayout: Multi-grained multimodal transformer for document understanding. *arXiv preprint arXiv:2209.08569*.

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.

Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, and Furu Wei. 2021. Layoutxlm: Multimodal pre-training for multilingual visually-rich document understanding. *arXiv preprint arXiv:2104.08836*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

A Appendix

A.1 Experiment Dataset

The data distribution and labeling of the dataset are shown in Table 5.

Table 5: Detail Data of Datasets. The nonmeaning label "OTHER" is not included.

Dataset	Label Num	Train	Dev	Test
FUNSD	3	149	-	50
CORD	30	800	100	100
SROIE	4	626	347	
XFUND	3	149	-	50

A.2 Hyper-parameters Setting

We show the training hyper-parameters on each dataset in Table 6.

Table 6: Finetuning Hyper-parameters. L, M, B and G refer to learning rate, max steps, batch size and gradient accumulation steps.

Dataset	Model size	Language	L	M	B	G
FUNSD	BASE	English	1e-5	2000	4	1
	LARGE		1e-5	2000	4	1
CORD	BASE	English	5e-5	2000	4	1
	LARGE		5e-5	3000	4	1
SROIE	BASE	English	1e-5	2000	2	1
	LARGE		1e-5	2000	8	1
XFUND	BASE	CHINESE	7e-5	2000	8	4