L3IE: Low-Resource Layout-Aware Information Extraction from Long Documents

Anonymous EMNLP submission

Abstract

Extracting information from full documents is an important problem in realistic applications, in which many documents processed in the information extraction pipeline are visually rich. However, most of the existing approaches typically take plain text as input, ignoring the rich layout information in the document. In this paper, we propose a practical layout-aware information extraction benchmark, L3IE, to facilitate the study on extracting both structural and semantic knowledge from long documents in a low resource scenario. Specifically, three tasks are released, including document structure extraction, attribute value extraction, and entity relation extraction. Each document in L3IE is parsed into a token sequence with corresponding layout information, and only a few documents are human-annotated for downstream tasks. We also provide baselines for the benchmark by incorporating pre-trained language models with layout information. Experimental results demonstrate that the performance can be significantly improved by equipping token-based models with layout awareness, and L3IE is still challenging for existing models.

1 Introduction

Information extraction (IE), defined as the task of identifying interested classes of entities as well as relations among these entities from unstructured or semi-structured documents, plays an important role in many knowledge-driven downstream tasks, including question answering (Mitra et al., 2019), dialogue system (Wu et al., 2020) and knowledge base population (Tanon et al., 2020). Most of the existing IE approaches to date assume the input to be text strings, while real-world IE systems often have to process business documents with rich visual layout, as the examples shown in Figure 1. In the process of converting such documents into plain text, a lot of layout information and visual features are lost, which is exactly the indicative signal that helps humans quickly understand document themes and search interested knowledge.

Recently, using layout information to help understand documents and extract information has attracted significant research interests. For example, SROIE (Huang et al., 2019) aims to extract company, date, address, or total from scanned receipts. Similarly, FUNSD (Jaume et al., 2019) assigns to each word a semantic entity label from a set of four predefined categories, including question, answer, header, and other. However, these benchmarks only focus on extracting unary spans from single-page documents, ignoring the extraction of fact knowledge, the relational triplets in the form of (*subject*, *relation*, *object*), from long documents. Besides, the state-of-the-art document understanding models on the above benchmarks, LayoutLM (Xu et al., 2020b) and LayoutLMv2 (Xu et al., 2020a), collect a large number of layout-aligned corpus (more than 11 million scanned document pages) to pre-train layout-aware language models, which is quite expensive to apply such methods to new practical applications in other languages or domains, especially the scene with low document resource.

In this paper, we propose a new document-level information extraction benchmark L3IE[1] with limited visually rich documents to exploit more practical IE systems. Specifically, given a small amount of annotated data (20 documents) and a relatively large amount of unannotated in-domain data (3000 documents), we first collect all tokens with their 2-d positions within the document, and further design three IE tasks in such low resource scenario: *document structure extraction*, *attribute value extraction* and *entity relation extraction*. The first task focuses on extracting the structural knowledge of long visually rich documents, while the latter two tasks measure the ability of the model to extract semantic fact knowledge under the open-world or

[1] L3IE is the abbreviation of Low-resource Layout-aware Information Extracion from Long documents.

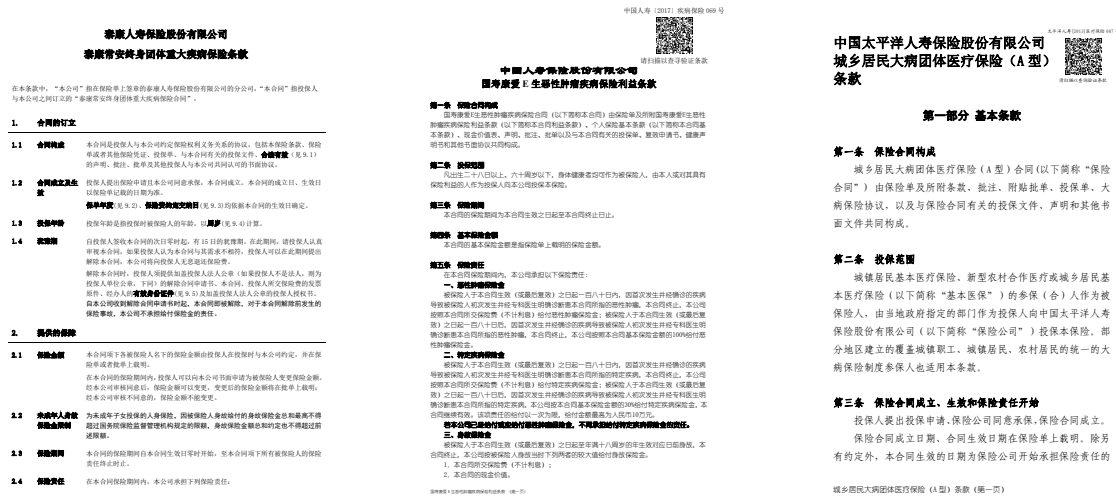


Figure 1: Real-world business documents with different layouts and formats.

closed-world assumptions (Fader et al., 2011).

In order to train layout-aware IE models with low document resources, we propose to start with an pre-trained language model (e.g., BERT (Devlin et al., 2019)), and fine-tune the model to endow it the ability to perceive document layout. Inspired by the BERT model, we introduce layout embedding in the fine-tuning phase, which describes the spatial position of a token within the document and is expected to be able to capture the relationship among tokens. Specifically, we generate four layout embeddings for a token-based on its four coordinates (x_0, y_0, x_1, y_1), where (x_0, y_0) corresponds to the upper-left position of the token and (x_1, y_1) represents the lower-right position. We then design three strategies to incorporate layout embeddings into the input layer, attention layer, output layer of transformer-based language models, respectively. With only a few training data in the supervised fine-tuning stage, the newly introduced layout embedding tables may not be thoroughly optimized. We propose to first fine-tune the token-based pre-trained language models on unlabeled in-domain data with unsupervised training objective, and then fine-tune the layout-aware models on labeled data for each specific task.

Experimental results suggest that injecting layout knowledge into pre-trained language models significantly improves the performance of layout-aware IE, and the unsupervised fine-tuning process with unlabeled in-domain data dramatically accelerates the convergence of the models on specific downstream tasks. Nevertheless, the overall performance is not satisfactory for real-world applications, demonstrating that the proposed L3IE benchmark is still challenging for existing models. Through comprehensive analysis, we also highlight several possible future directions for the low-resource layout-aware information extraction problem. We release the benchmark and baselines to the community[2] for future research.

2 Related Work

Extracting information from visually rich documents has been studied for decades. Before the advent of learning-based models, many rule-based methods were proposed (Riloff, 1993; Muslea, 1999; Belaïd et al., 2001; Rusinol et al., 2013). Though these methods work in some cases and do not rely heavily on annotated data, the design and maintenance of rules require deep expertise and large time cost, they cannot generalize across document templates, Lately, LayoutLM (Xu et al., 2020b) and LayoutLMv2 (Xu et al., 2020a) are proposed to jointly model interactions between text and layout information across scanned document images, which are beneficial for some document understanding tasks like IE from visually rich documents. They are pre-trained on the IIT-CDIP Test Collection 1.0 (Lewis et al., 2006), which contains more than 11 million scanned document pages and each document exactly has its corresponding text and metadata. The high quality and quantity of data brings superior performance to these models, but

[2] https://github.com/xxx/xxx

limits their transferability since not every language or field is enough to collect such data in practical applications. Beyond that, Yang et al. (2017) treated the document semantic structure extraction task as a pixel-by-pixel classification problem and proposed a multimodal neural network that considers visual and textual information. Zhang et al. (2020) presented an end-to-end network to bridge text reading and IE for document understanding. Wei et al. (2020) combined the power of large pre-trained language models and graph neural networks for the structural-aware extraction from visually-rich documents. However, it needs sufficient annotation data in the training stage, and cannot work well with limited annotation resources.

Recently, more and more datasets for layout-aware IE have been proposed. CORD (Park et al., 2019) includes totally 1000 receipts, where a photo with a list of OCR annotations are equipped for each receipt. The dataset defines 30 fields under 4 categories (*info*, *menu*, *total*, *etc.*) and the task aims to label each word to the right field. SROIE (Huang et al., 2019) is a public dataset with 973 receipts in ICDAR 2019 challenge, which aims to extract values from each receipt of up to 4 pre-defined keys: *company*, *date*, *address*, *total*. FUNSD (Jaume et al., 2019) is another dataset with 199 fully annotated forms, and the most popular task is to assign to each word a semantic entity label from 4 categories: *question*, *answer*, *header* or *other*. PubLayNet (Zhong et al., 2019) is a large document layout annotation dataset with $360k$ document images, and the task is to identify *text*, *titles*, *tables*, *lists* and *figures* from the document. DocBank (Li et al., 2020) contains $500k$ document pages with token-level annotations for the layout recognition of scientific articles, in which there are totally 12 fine-grained categories such as *title*, *abstract*, *section*, *paragraph*. Besides, the InvoiceIE and ResumeIE datasets (Wei et al., 2020) are also formulated as two entity extraction tasks technically, in which the former aims to extract 4 types of entities (*amount*, *seller_name*, *purchaser_name*, and *invoice_no.*) from invoices, while the later aims to extract 11 types of entities such as *name*, *school* and *phone_number* from resumes. Overall, almost all of these datasets are solely focused on the extraction of a few types of entities from short documents, usually one page. It limits the research of relation extraction tasks from long documents, which is a more challenging practical requirement.

3 The L3IE Benchmark

Compared with existing layout-aware IE datasets, L3IE is formed with several desirable properties: (i) It is proposed for the low resource scenario, which is more challenging and practical than traditional data-abundant settings. (ii) It pays more attention to the extraction of fact knowledge, which could be directly applied to the knowledge base construction and facilitate knowledge-driven downstream tasks. (iii) It needs to extract information from long documents, which is more suitable for real application scenarios. In this section, we first introduce the task formulation, then detail the collection of human-annotated data, and analyze various aspects of L3IE to provide a deeper understanding of the benchmark. Finally, we provide baseline models for each task with only token-based input.

3.1 Task Formulation

To comprehensively evaluate the extraction ability of the model from long visually-rich documents, we formulate three sub-tasks for L3IE: *document structure extraction*, *attribute value extraction* and *entity relation extraction*.

Document Structure Extraction. It aims at parsing the hierarchical structure of section headings for the input document. In other words, the goal is to detect section *headings* with its *levels* and generate corresponding *catalog numbers*. For instance, in Figure 2, "保险责任 (Insurance liability)" is the section heading with level 1 and number 5, and "恶性肿瘤保险金 (Cancer insurance)" is the subsection heading of "保险责任 (Insurance liability)" with level 2 and the number 5.1. The task is important for applications such as document retrieval and document question answering (Lewis et al., 2020).

Attribute Value Extraction. It can be regarded as a special form of Open IE, which does not rely on pre-defined ontology schema (Mausam, 2016). An Open IE system converts natural text to semi-structured representations, by extracting a set of relational facts organized as triples in the form of (*subject*, *attribute*, *value*), where each element is a continuous span in the input text. Different from Open IE, attribute value extraction sets up a more focused extraction direction, that is extracting tuples organized as (*attribute*, *value*) from the input document towards a target *subject* (typically the title of the input document). The second line of Figure 2 shows a concrete example.

Entity Relation Extraction. It is a well-defined

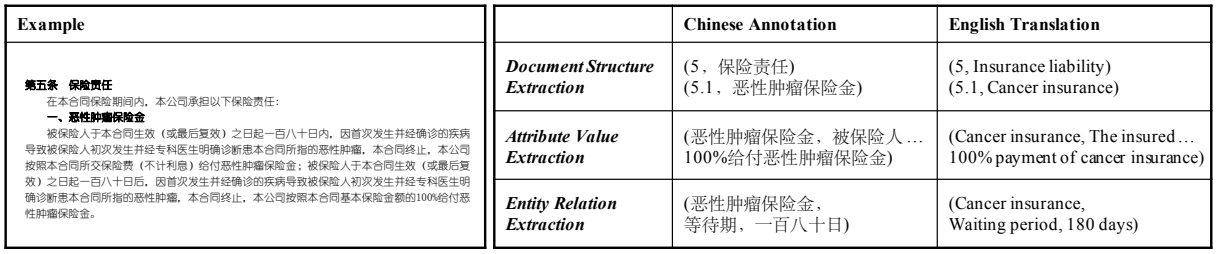


Figure 2: A snapshot from L3IE with demonstrations of the proposed three tasks.

task to detect two *entities* and recognize their semantic *relations* from the document (Zheng et al., 2017), as the third row of Figure 2 shows. Different from attribute value extraction whose attribute (relation) words are extracted from documents, in this task, relations are predicted from a predefined set with 18 relations (see also appendix). Thus, it is suitable for obtaining structured knowledge with specific relations that we care about.

3.2 Data Collection and Labeling

Basically, we download 3100 business documents in PDF format from the website of Insurance Association of China[3] to create the L3IE benchmark. These PDF files come from 133 companies and cover 17 common product categories in the insurance industry. The documents of a company typically have one or two layout forms, which makes the dataset more diverse and robust to real applications. We focus on Chinese documents in this work and will expand to other languages in the future.

For the generation of layout information, we use PDFPlumber[4], a PDF parser built on PDFMiner[5], to extract tokens with their bounding boxes. To represent the spatial position of tokens, a document page is considered as a coordinate system with the top-left origin, and the bounding boxes are defined as the most upper-left and lower-right coordinates of tokens. Formally, we utilize (x_0, y_0) to represent the position of the upper-left in the bounding box, and (x_1, y_1) to represent the position of the lower-right. It means that each word corresponds to four coordinates in the parsing results.

Next, we randomly select 100 documents from 5 companies and annotate each of them for all three tasks. The process of data annotation is carried out by crowdsourcing. Annotators are provided with the PDF files, from which they need to copy text spans or generate numbers and fill them into corresponding slots (e.g., heading and catalog number) in result files. Every annotator needs to be trained and tested for specific tasks, and only after passing the test can he/she start the formal work. The annotation results are also randomly checked. Once the error rate of a document exceeds 5%, all the annotations of the annotator will be reviewed.

Finally, we align the parsing results with annotation results to ensure that all annotated fields can be found in the parsed document. If the annotated field cannot found, we try to match the start and end substrings of the annotated field and locate the text segment with the minimum edit distance from the annotation. In the end, there are still ∼3% of the annotation instances that cannot be located, we discard them and impute them to parsing errors.

3.3 Dataset Statistics

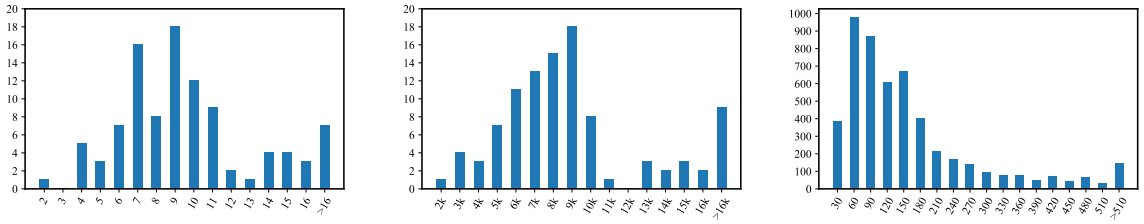
The final L3IE benchmark consists of 100 human-annotated business documents and 3000 unannotated in-domain documents, all of which have parallel layout information. We analyze the 100 annotated documents in detail to support the follow-up study. Table 1 provides basic statistics and a comparison of L3IE to the previous layout-aware document understanding datasets. The total document pages of L3IE is 1053, which is the second-highest among the compared datasets. Meanwhile, all the listed datasets are single-page based and aim to extract unary spans, while only L3IE considers the full document with consecutive pages and supports the extraction of fact knowledge.

Furthermore, we show the distributions of page and word numbers in one document in Figure 3(a) and 3(b), respectively. The average page number is 10.53 and the average word number is 10469. Figure 3(c) also depict the distribution of value length, from which one can observe that the value length is concentrated between 60 and 90, and some even length exceeds 510. All of these are beyond the fo-

[3] http://www.iachina.cn/
[4] https://github.com/jsvine/pdfplumber
[5] https://github.com/euske/pdfminer

Datasets	#Documents	#Pages	Consecutive?	Span Extraction?	Fact Extraction?
CORD (Park et al., 2019)	1000	1000	✗	✓	✗
SROIE (Huang et al., 2019)	973	973	✗	✓	✗
FUNSD (Jaume et al., 2019)	199	199	✗	✓	✗
DocBank (Li et al., 2020)	-	500000	✗	✓	✗
L3IE (annotated)	100	1053	✓	✓	✓

Table 1: Comparison of L3IE with representative layout-aware document understanding benchmarks.



(a) Distribution of document pages. (b) Distribution of document length. (c) Distribution of value length.

Figure 3: Distributions of document pages, document length (word number) and value length in L3IE.

	Train	Dev	Test
Document Structure Extraction	1203	2452	2420
Attribute Value Extraction	969	2036	2092
Entity Relation Extraction	340	691	672

Table 2: Statistics of the train/dev/test datasets.

cus of most existing IE benchmarks and pose new challenges to the document modeling process. In order to maximize the reusability of the benchmark, we provide a predefined split of the annotated data into train, dev, and test sets and report the statistics in Table 2. The train set is quite smaller than the dev and test sets, which simulates the low resource scenario and is very rare in existing datasets.

3.4 Baselines

In order to comprehensively evaluate the challenges of L3IE, we propose a strong set of baselines. The design philosophy for baseline models is trying to ensure their simplicity. For this purpose, we summarize the three tasks into a unified sequence labeling framework with different tagging schemes to describe the features of each word.

Document Structure Extraction. The model labels the heading spans as well as their level in the input document. In L3IE, the heading has three levels from 1 to 3: section, subsection, and subsubsection. So the tag set is defined as {`B-L1`, `B-L2`, `B-L3`, `E-L1`, `E-L2`, `E-L3`, `O`}, in which `B` and `E` denote where each word locating in the beginning or end index of one heading, and `L1-L3` encodes its level. Tag `O` represents the other tag, which means that the corresponding word is independent of the extracted results. During decoding, we can obtain each heading span by the way of boundary nearest matching, and then number it in order according to the level and position of the heading, so as to determine the document structure.

Attribute Value Extraction. The model labels the span of attribute and value with the tag set {`B-V`, `B-A`, `E-V`, `E-A`, `O`}. In the decoding process, we combine every two attribute and value into a tuple based on the nearest principle.

Entity Relation Extraction. The model is more complex than that in attribute value extraction as it not only extracts two eneities and matches them, but also determines the semantic relation between them. We design a tri-part tagging scheme {`B-S`, `E-S`, `B-O-Rel`, `E-O-Rel`, `O`}, where `S` and `O` denote subject and object respectively, `Rel`$\in \mathcal{R}$ denotes predefined relation. Thus, the total number of tags is $2 \times |\mathcal{R}| + 3$. During decoding, we first detect the subject entity, then the object entity and its involved relation from the object tags, and finally pair them according to the nearest principle.

4 Methodology

To take advantage of large transformer-based language models and adapt to layout-aware IE tasks, we start from existing pre-trained models and inject layout features into them. Unlike 1-d position embedding which models the token position in a sequence, we introduce 2-d layout embedding to capture the spatial location of a token in the document.

We normalize and discretize all coordinates to integers in the range $[0, 100]$ and look-up layout embeddings $\{\mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_1, \mathbf{y}_1\}$ from four embedding tables. Next, we add them up $\mathbf{l} = \mathbf{x}_0 + \mathbf{y}_0 + \mathbf{x}_1 + \mathbf{y}_1$ as the final layout embedding and propose three strategies to incorporate layout features with transformer-based language models in the input layer, attention layer and output layer as follows.

Layout-Aware Input Layer. Following the popular choice of incorporating position embedding in transformer-based language models (Devlin et al., 2019), for each token t_i, we add the layout embeddings $\mathbf{l}_i$ with text embeddings $\mathbf{t}_i$ to achieve layout-aware input embeddings $\mathbf{x}_i = \mathbf{t}_i + \mathbf{l}_i$. We hope that the model can automatically perceive the layout information in the following learning process.

Layout-Aware Attention Layer. To inject layout features into the inner structure of the transformer, we introduce a layout-aware self-attention mechanism by considering layout information in the calculation process of attention score. The vanilla self-attention mechanism captures the correlation between query $\mathbf{x}_i$ and key $\mathbf{x}_j$ by projecting these two vectors and calculating the attention score $a_{ij} = \frac{1}{\sqrt{d_{head}}}(\mathbf{x}_i\mathbf{W}_q)(\mathbf{x}_j\mathbf{W}_k)^\top$. Here we perform a similar calculation for layout embeddings and explicitly add the result to the original attention score $a'_{ij} = a_{ij} + \frac{1}{\sqrt{d_{head}}}(\mathbf{l}_i\mathbf{W}'_q)(\mathbf{l}_j\mathbf{W}'_k)^\top$.

Layout-Aware Output Layer. In the output layer, we concatenate the hidden representation $\mathbf{h}_i$ of the last layer of transformer with layout information $\mathbf{o}_i = [\mathbf{h}_i; \mathbf{l}_i]$. This is also a common solution for information aggregation (Yao et al., 2019).

However, labeling plentiful data is expensive for document-level IE applications, especially for the low resource domains, but it is relatively easy to obtain some unlabeled data related to downstream tasks. Inspired by existing pre-trained models, we propose to fine-tune on unlabeled in-domain data before supervised training. Similar with the objective of masked language model, we propose layout-aware masked language model (LMLM) to learn language representation with both layout and text embeddings. During the fine-tuning, we randomly mask some of the input tokens but keep their corresponding layout embeddings, then the model is trained to predict the masked tokens given the context and 2-d position. In this way, the model not only understands the language contexts but also utilizes the layout information, thereby the token-based model is equipped with layout awareness.

5 Experiments

In this section, we conduct comprehensive experiments to evaluate different layout-aware models on the benchmark. Through detailed analysis, we also discuss several future directions for L3IE.

5.1 Experimental Setup

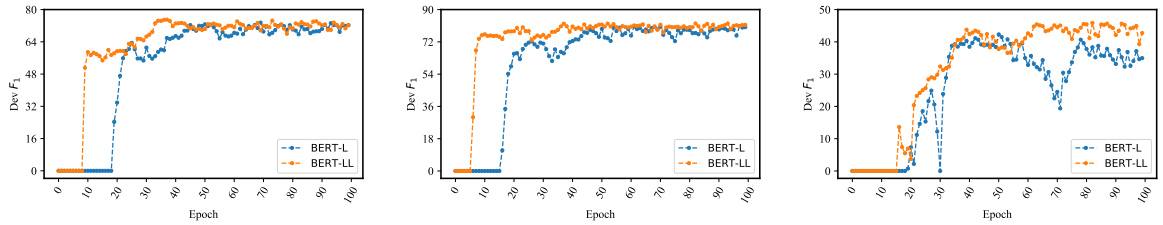
Implementation Details. We employ BERT (Devlin et al., 2019) and Longformer (Beltagy et al., 2020) as the backbone pre-trained language models, and implement our model based on Transformers (Wolf et al., 2020) and use the official *bert-base-chinese* model and the *schen/longformer-chinese-base-4096*[6] model uploaded by developer. The input sequence is trimmed to a maximum length of 512 and 4096, respectively. The size of layout embedding is set to 768. The learning rates of fine tuning are both set to $1e^{-5}$ (chosen from $1e^{-3}$ to $1e^{-6}$). We optimize our model with Adam and run it on one 32G Tesla V100 GPU for 100 epochs. All hyper-parameters are tuned on the dev set.

Evaluation Metrics. As the standard-setting in IE tasks, we adopt micro-average F1 score, the harmonic mean of precision and recall, for evaluation. Formally, the precision and recall are defined as $P = S_m/|\mathcal{P}|$, $R = S_m/|\mathcal{G}|$, in which S_m is the matching score of all predicted results, $|\mathcal{P}|$ and $|\mathcal{G}|$ denotes the size of prediction set and ground truth set, respectively. For document structure extraction and entity relation extraction, $S_m = \sum_i^{|\mathcal{P}|} m(p_i)$, where $m(p_i)$ is set to 1 if the i-th predicted result matches the ground truth strictly, otherwise it is 0. In document structure extraction, we use both level-based metrics and catalog-based metrics, where the former measures the performance of tuple (*level*, *heading*) and the later measures (*catalog number*, *heading*). For attribute value extraction, we calculate S_m, as the evaluation in Open IE systems, by computing the similarity between each predicted fact in $\mathcal{P}$ and each ground truth fact in $\mathcal{G}$, then find the optimal matching to maximize the sum of matched similarities by solving a linear assignment problem (Sun et al., 2018). In the procedure, the similarity between two facts is defined as $s(p_i, g_j) = \sum_{l=1}^{2} \mathcal{M}(p_i^x, g_j^x)/2$, where p_i^x and g_j^x denote the x-th element (i.e., *attribute* or *value*) of fact p_i and g_j respectively, $\mathcal{M}(\cdot,\cdot)$ denotes the gestalt pattern matching measure (Ratcliff and Metzener, 1988) for two strings.

[6] https://huggingface.co/schen/longformer-chinese-base-4096

Models	Doc. Struc. Ext.			Attr. Val. Ext.			Ent. Rel. Ext.		
	P	R	F1	P	R	F1	P	R	F1
BERT	80.7	65.7	72.4	75.8	80.1	78.1	46.1	39.1	42.3
BERT-L (+Layout)	73.3	73.8	73.6	80.5	80.2	80.3	48.6	38.7	43.1
BERT-LL (+Layout+LMLM)	80.9	69.9	**75.0**	77.6	87.7	**82.3**	49.5	42.7	**45.9**
Longformer	75.4	67.6	71.3	75.5	71.1	73.2	35.3	37.3	36.3
Longformer-L (+Layout)	75.4	70.9	73.1	74.3	73.4	73.8	34.6	16.9	22.7
Longformer-LL (+Layout+LMLM)	76.0	77.9	**76.9**	84.1	82.2	**83.2**	51.7	39.3	**44.6**

Table 3: Main results of different models on L3IE. We adopt the layout-aware input layer strategy in fine-tuning.



(a) Document structure extraction. (b) Attribute value extraction. (c) Entity relation extraction.

Figure 4: Convergence processes of layout-aware models on L3IE with and without unsupervised fine-tuning.

Models	Doc. Struc. Ext.		
	P	R	F1
BERT	19.8	16.1	17.8
BERT + Layout	26.2	21.6	23.7
BERT + Layout + LMLM	29.8	23.7	26.4
Longformer	17.5	16.1	16.8
Longformer + Layout	20.9	21.3	21.1
Longformer + Layout + LMLM	30.6	31.3	31.0

Table 4: Results of document structure extraction with catalog-based metrics. It measures the performance of the predicted (*catalog number*, *headings*) tuples.

5.2 Main Results

Comparing the performance of different models in Table 3, the first conclusion we draw is that layout-aware models outperform vanilla token-based models in almost all the evaluation matrices, which demonstrates the effectiveness of incorporating layout information with pre-trained language models. Secondly, the preliminary process of in-domain unsupervised fine-tuning further improves the performance. Through further analysis, we also find that unsupervised fine-tuning dramatically accelerates the convergence in all downstream tasks (see Figure 4). Thirdly, the replacement of BERT with longformer does not bring expected noticeable improvement, and the performances have been greatly improved with unsupervised fine-tuning. We attribute it to the pre-training process since there is only the pre-trained model but no description of the pre-training corpus and implementation details. In this experiment, we are more interested in the improvement of layout and leave the pre-training of long-form language models as future work.

Furthermore, when looking at the results in different tasks, we find that: (1) Attention value extraction holds the highest performance, it is intuitive since the task has the smallest number of tags and is simplest for the sequence labeling solutions. (2) The performance of document structure extraction takes second place. However, in Table 4, the results with catalog-based metrics are near collapse, which indicates that there is a long way to go on this task, the main error is also discussed in Section 5.4. (3) All of the models perform worst in entity relation extraction, we analyze the reason behind it is that there are the minimum facts but the maximum tags, which is difficult for optimization. Thus, we also highlight pertinently mining task-related layout information as a future direction.

5.3 Performance Analysis

To gain more insight into the characteristics of different layout-aware models, we perform detailed analyses from two aspects. Here we select BERT as the backbone pre-trained model and the results of longformer have similar tendencies.

Different Strategies of Integrating Layout. Table 5 summarizes the results on all tasks with different layout-aware strategies. From the results, we find that incorporating layout information in

Models	Doc. Struc. Ext.			Attr. Val. Ext.			Ent. Rel. Ext.		
	P	R	F1	P	R	F1	P	R	F1
BERT-LL (input)	80.9	69.9	75.0	77.6	87.7	**82.3**	49.5	42.7	**45.9**
BERT-LL (attention)	79.9	72.4	**76.0**	76.8	81.0	78.8	43.7	51.9	37.7
BERT-LL (output)	81.6	69.5	75.1	82.1	76.4	79.2	51.9	39.1	44.6

Table 5: Results on L3IE with different strategies of integrating layout information.

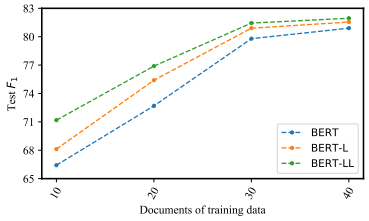


Figure 5: Results of document structure extraction with different amount of training data.

the input layer broadly outperforms the other two strategies. It follows the customs of integrating position knowledge into transformer-based language models and provides the possibility to benefit from the following architecture to model layout information. Moreover, not all the layout-aware methods bring noticeable improvement to the token-based model, so designing a more reasonable layout integration method based on transformer is also a point that needs to be studied in the future.

Difference Amounts of Training Data. Here we explore the performances on specific downstream tasks with different amounts of supervised training data. We reorganize the 60 documents in the train and dev sets and keep the test set unchanged for a fair comparison. In this experiment, we take {10, 20, 30, 40} documents as new train set, and the rest as dev set. From the result in Figure 5, we find that the performance gap between BERT-L and BERT increases first and then decreases. We analyze that the newly introduced layout embedding tables cannot be fully optimized with only a small amount of data, which limits the performance of BERT-L, when the training data increases to enough, BERT is also capable of capturing some semantic information related to decision-making. On the contrary, the unsupervised fine-tuning stage reduces the sensitivity of the amount of training data and achieves better performances, especially when there are only a few annotation data.

5.4 Error Analysis and Limitations

We analyze the outputs on the dev set and find that there are three typical errors for current models: (1) In document structure extraction, the predicted results miss some section headings, for example, the model outputs the first and third section headings but misses the second heading, resulting in a continuous error in catalog-based metrics after the second heading. One possible solution is to introduce fine-tuning objectives (e.g., relative position prediction) to model elements with the same alignment format. (2) In attribute value extraction, the errors are mainly lie in the prediction of attribute items. Note that many attributes are subsection headings, it might be useful to consider the document structure in this task or jointly model attribute value extraction and document structure extraction with a multi-task learning framework. (3) In entity relation extraction, there are some cases that the subject and object entities are correctly predicted but mismatched, since not all of the triple facts are in line with the nearest principal. Therefore, it is necessary to design more reasonable tagging and decoding schemas in the future.

A major limitation in L3IE is that all documents for annotation are from the insurance industry, which may limit the diversity and generality, especially for the relation distributions in entity relation extraction. It might be helpful to extend L3IE with documents from other fields, such as financial reports and electronic medical records, to design a more comprehensive relation schema.

6 Conclusions

In this paper, we present the first low resource layout-aware information extraction benchmark, which is designed for the extraction of structural and semantic knowledge from long visually rich documents. We also investigate the performance of incorporating pre-trained language models with layout information by using labeled and unlabeled data. Experimental results demonstrate the effectiveness and highlight several future directions: (1) developing more advanced token-based language models, (2) exploring more effective layout integration strategies, and (3) designing more reasonable solutions for specific downstream tasks.

Ethical Consideration

This work does not present any direct societal consequence. The proposed work seeks to present a new benchmark for layout-aware document-level information extraction with low resources, especially for the extraction of fact knowledge from long documents. In the crowdsourcing process, we offer annotators $14 for each document, in which $7 for the document structure extraction and attribute value extraction tasks, $7 for the entity relation extraction task. We believe this leads to practical information extraction systems that benefit the information extraction community where annotating massive document-level training data is expensive and immensely limits the transferability. And it potentially has broad impacts since the tackled issues also widely exist in tasks of other areas.

References

Abdel Belaïd, Yolande Belaïd, Late N Valverde, and Saddok Kebairi. 2001. Adaptive technology for mail-order form segmentation. In *Proceedings of Sixth International Conference on Document Analysis and Recognition*, pages 689–693. IEEE.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition*, pages 1516–1520.

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*, pages 1–6.

David Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David Grossman, and Jefferson Heard. 2006. Building a test collection for complex document information processing. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 665–666.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. DocBank: A benchmark dataset for document layout analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 4074–4077. IJCAI/AAAI Press.

Arindam Mitra, Peter Clark, Oyvind Tafjord, and Chitta Baral. 2019. Declarative question answering over knowledge bases containing natural language text with answer set programming. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 3003–3010. AAAI Press.

Ion Muslea. 1999. Extraction patterns for information extraction tasks: A survey.

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*.

John W Ratcliff and David E Metzener. 1988. Pattern-matching-the gestalt approach. *Dr Dobbs Journal*, 13(7):46.

Ellen Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In *AAAI*, volume 1, pages 2–1. Citeseer.

Marçal Rusinol, Tayeb Benkhelfallah, and Vincent Poulain dAndecy. 2013. Field extraction from administrative documents by incremental structural

templates. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1100–1104. IEEE.

Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. 2018. Logician: A unified end-to-end neural approach for open-domain information extraction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 556–564. ACM.

Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. 2020. Yago 4: A reason-able knowledge base. In *European Semantic Web Conference*, pages 583–596. Springer.

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. ACM.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sixing Wu, Ying Li, Dawei Zhang, Yang Zhou, and Zhonghai Wu. 2020. Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5811–5820, Online. Association for Computational Linguistics.

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 *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C. Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4342–4351. IEEE Computer Society.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. 2020. TRIE: end-to-end text reading and information extraction for document understanding. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pages 1413–1422.

Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. 2017. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59–66.

Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition*, pages 1015–1022. IEEE.