Language, OCR, Form Independent (LOFI) pipeline for Industrial Document Information Extraction

Chang Oh Yoon[1,+], Wonbeen Lee[1,+], Seokhwan Jang[1,+], Kyuwon Choi[2,+], Minsung Jung[2,+], Daewoo Choi[3,*],

[+]AgileSoDA, [*]Hankuk University of Foreign Studies

[+]{coyoon,wblee,jsh,,kwchoi,timmy92}@agilesoda.ai,[*]daewoo.choi@hufs.ac.kr

Abstract

This paper presents LOFI (Language, OCR, Form Independent), a pipeline for Document Information Extraction (DIE) in Low-Resource Language (LRL) business documents. LOFI pipeline solves language, Optical Character Recognition (OCR), and form dependencies through flexible model architecture, a token-level box split algorithm, and the SPADE decoder. Experiments on Korean and Japanese documents demonstrate high performance in Semantic Entity Recognition (SER) task without additional pre-training. The pipeline's effectiveness is validated through real-world applications in insurance and tax-free declaration services, advancing DIE capabilities for diverse languages and document types in industrial settings.

1 Introduction

Many industries handle complex documents known as Visually Rich Documents (VRDs), containing text, tables, and figures. In real-world industry scenarios involving VRDs, we should consider a process of Semantic Entity Recognition (SER) (Cui et al., 2021) to automate workflows. For example, in insurance claims processing, patient information and diagnostic details need to be extracted from medical reports submitted by customers. Additionally, in accounting and tax filing processes, purchase information should be extracted from receipts or other tax documents.

To address the automation demands of the industry, we face three main challenges:

1. There are no publicly available VRD datasets in Low-Resource Languages (LRL), which makes it difficult to create pretrained models, nor are there any publicly available models for these languages.

2. There are limitations in SER from OCR engine results. Typically, OCR engine results are at the word level, but those OCR results often require extra splitting or combining to get semantic entities.

3. Documents handled in the industry also present challenges in information extraction due to custom formats, even when standardized forms exist. For example, in medical reports, even though there is a standardized form mandated by the government, some hospitals use their own custom formats. Similarly, receipts may contain simple information, but their format varies significantly across institutions. Regardless of the document type, rotation or distortion of images can also change the document's structure.

However, related research has not comprehensively addressed these three issues together. We have focused on considering these three challenges collectively in order to meet the automation demands of the industry.

Language Independence: There's a lack of publicly available datasets and models that work with LRL, languages that are less used compared to English and Chinese, such as Korean and Japanese. Most VRD datasets, such as EPHOIE, FUNSD, and CORD (Wang et al., 2021; Jaume et al., 2019; Park et al., 2019) are primarily in English or Chinese, and most open models (LayoutLM, LayoutLMv2, LayoutLMv3, BROS, GeoLayoutLM) (Xu et al., 2020b,a; Huang et al., 2022; Hong et al., 2022; Luo et al., 2023) are trained with open datasets (Lewis et al., 2006). As multilingual models like LayoutXLM (Xu et al., 2021) and LiLT (Wang et al., 2022) exist, we choose LiLT for our base model due to its flexibility across different languages.

OCR Independence: Models like LayoutLM, LayoutLMv2, LayoutLMv3, LiLT, BROS, and GeoLayoutLM use word-level or segment-level bounding boxes to encode spatial information of text. However, languages with linguistic features

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1056–1067

differing from English face challenges in extracting such bounding boxes. For instance, Japanese lacks spaces between words (Tian et al., 2020; Higashiyama et al., 2022), and Korean employs particles (Seo et al., 2023), resulting in single bounding boxes containing multiple words with distinct semantic meanings. Consequently, the complexity of bounding boxes varies across languages. Therefore, a framework capable of performing SER independent of any OCR engine result used is essential. See Figure 5 in Appendix for an example of token-level box split algorithm.

Form Independence: To create the model's input format from document images, the OCR engine results need to be arranged in an appropriate reading order. However, documents that occur in real industries are mostly photos, faxes, scanned copies, etc., which frequently have distortions or rotations (Chen et al., 2024). For documents with these characteristics or complex forms, it is difficult to determine the appropriate reading order (Wang et al., 2023).

In this paper, we present a practical DIE pipeline for SER tasks, LOFI (Language, OCR, Form independent Extraction) pipeline. Our experiments on Korean medical bills and Japanese receipts demonstrate its effectiveness, achieving entity-level F1 scores of 95.64% and 94.60%, respectively. Our main contributions are:

- A flexible pipeline structure that accounts for multiple factors in industrial DIE.
- Empirical evidence of satisfactory performance on Korean and Japanese industrial documents without additional pre-training.

2 Related Works

In this section, we show related works on language, OCR, and form methodologies on Document Information Extraction (DIE) on Semantic Entity Recognition (SER) tasks.

2.1 Language-independent Layout Transformer

The development of pre-trained DIE models for Low-Resource Languages (LRL) presents significant challenges. Acquiring enough LRL documents for pre-training is a time-consuming and arduous task (Wang et al., 2022), which is added by the scarcity of publicly available LRL documents.

The LiLT model has a structure that can address these challenges. LiLT discovered that among the text and layout, called bounding boxes, crucial in DIE tasks, layout is relatively language-independent (Wang et al., 2022). This allowed for handling non-English documents by changing the text encoder layers of a DIE model pre-trained on English documents to a multilingual Pre-trained Language Model (PLM) (Wang et al., 2022). This compatibility comes from the language-independent interaction between layout encoder layers and text encoder layers during computation, resulting in independent effects of layout and text. To handle LRL documents, we replace the text encoder layers in the LiLT model structure to a PLM for the respective language, enabling us to process LRL documents.

2.2 Representation of spatial information within documents

Models for DIE use text and its corresponding layout called bounding boxes as inputs. In real-world scenario documents, OCR engines are typically used to obtain text and bounding boxes. However, OCR engines may not provide the desired text and bounding boxes depending on the linguistic and structural characteristics of the document.

As mentioned in Introduction, Japanese documents lack spaces due to linguistic features, while Korean documents have particles, resulting in bounding boxes being extracted in various forms (character-level, word-level, line-level) (Kjøller Bjerregaard et al., 2022; Kim et al., 2022; Bryan et al., 2023). As such, when OCR engine results are extracted in such diverse forms, it causes performance degradation in the SER model that uses these results as input. VGT's approach to document layout analysis offers an alternative method (Da et al., 2023). This method uses a tokenizer to divide text into tokens, then equally splits the bounding boxes for each token and embeds it as a grid feature. However, VGT's uniform splitting of bounding boxes fails to reflect the actual length of tokens, which is a limitation. To address this limitation, we enhanced the algorithm.

This approach allows us to generate consistent token-level bounding boxes, independent of the OCR engine used. We've named this process the "Token-level box split" algorithm. This method preserves the technical integrity of DIE while addressing challenges posed by varying OCR engine results.

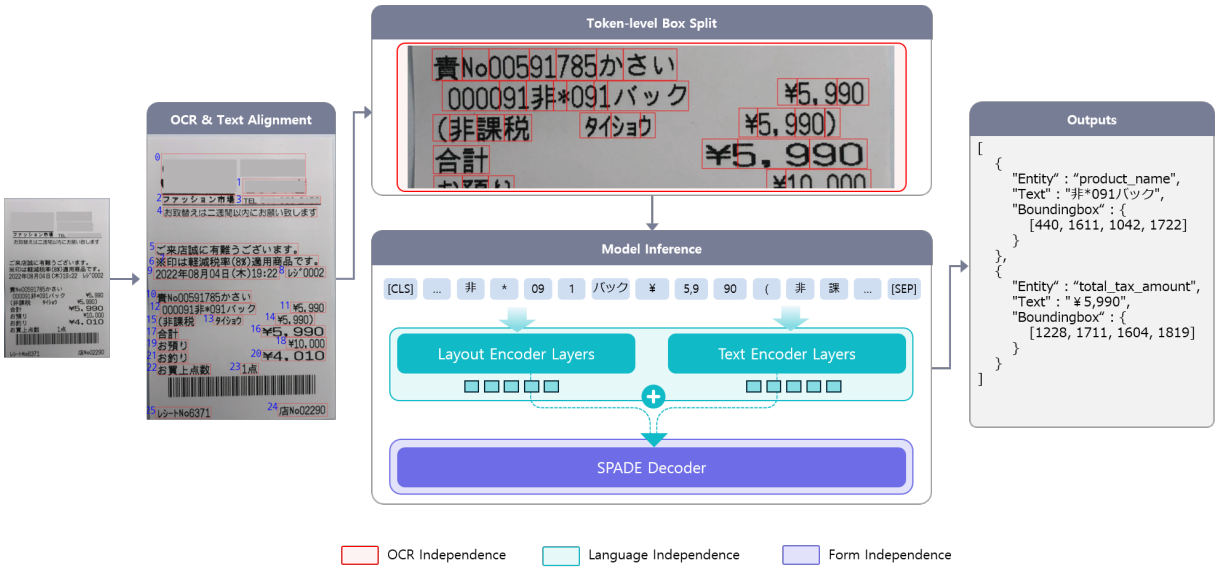


Figure 1: An example of LOFI Pipeline for a Japanese receipt. Language independence is solved using LiLT as the backbone model as shown as a teal box. OCR independence is solved using token-level box split algorithm as shown as a red box. Form independence is solved using SPADE decoder as shown as a purple box.

2.3 Graph based parser

In (Xu et al., 2020b,a; Huang et al., 2022; Wang et al., 2022), SER is performed using BIO tagging by applying token classification to the Transformer encoder. This method requires converting text and bounding boxes into a 1D input format compatible with transformer-based models. Consequently, the order of spatial information must be adjusted to align with entity units, enabling SER using BIO tagging (Zhang et al., 2023). However, in real-world scenarios, the numerous types of business documents used have diverse forms, limiting the ability to determine an appropriate reading order (Wang et al., 2023). This is mainly due to document features such as figures, tables, paragraphs, and font sizes. In particular, factors like document rotation, distortion, and noise also have an impact. To resolve these problems, we used a graph-based methodology, the SPADE decoder (Hong et al., 2022), in our pipeline.

3 LOFI Pipeline

In this section, we present the methodology of solving language, OCR, form dependency issues, and our LOFI (Language, OCR, Form Independent) pipeline, a DIE pipeline for SER tasks, as shown in Figure 1.

To outline LOFI pipeline process:

1. *OCR and text alignment*. Our own OCR engine generates text and bounding box data from document images. Then, to preprocess 1D positional information, the results are sequentially arranged from top-left to bottom-right.

2. *Token-level box split*. Our own algorithm is applied to the sorted text and bounding boxes, to preprocess 2D positional information.

3. *Model inference*. The (token, token box) pairs are put into LiLT for sequence output generation. The SPADE decoder processes this output to produce ITC and STC results.

4. *Outputs*. The results are combined to generate the final SER output.

The strengths of LOFI regarding the three challenges mentioned in Introduction are as follows.

Language Independence: Language models are paired with tokenizers, and Pretrained Language Models (PLMs) for specific languages typically use data predominantly in that language for tokenizer training. This ensures that tokens are structured to suit the characteristics of the language.

As discussed in Section 2.1, LiLT utilizes a model structure that can adapt to the PLM corresponding to the language of the target document, enabling customized token configurations for Low-Resource Languages (LRL).

In the teal box in Figure 1, we implement LiLT as the base model, utilizing a language-specific PLM for efficient token processing. Language-specific

models tokenize sentences into more contextually relevant tokens compared to multilingual models, which may be less optimal for single-language tasks and could suffer from parameter inefficiencies. Therefore, using an appropriate model enhances efficiency for our purposes.

OCR Independence: The model in our pipeline uses text and layout called bounding boxes as input. As mentioned in Section 2.2, the text and layout obtained through the OCR engine can have different ranges (character-level, word-level, line-level) depending on the linguistic and structural complexities of documents. This different range of bounding boxes results can lead to performance degradation in the SER model, as the range of layout is different if the OCR engines used in inference are different from those used in fine-tuning. For example, when using word-level bounding boxes for fine-tuning and line-level bounding boxes for inference, the layout ranges differently, which causes performance degradation.

We use the token-level box split algorithm to make the layout at the same level with any OCR engine. The algorithm converts any bounding box range (character-level, word-level, line-level) to the same token-level bounding boxes, which allows any OCR engine to be independent of the model's results. For details, refer to Algorithm 2 in the Appendix.

Form Independence: Regardless of the document format, OCR results need to be aligned for human-readable order. However, as mentioned in Section 2.3, this is a challenging task. Nevertheless, a consistent alignment is needed when constructing model inputs; a traditional method of Top-Left to Bottom-Right(TL-BR) alignment is used.

Figure 1's OCR & Text Alignment shows the text input order aligned in TL-BR. In the middle of the receipt, for items 13, 14, and 15, it fails to align in the correct order of 15→13→14 due to differences in bounding box positions. This is due to rotation and distortion characteristics occurring in real-world scenarios, along with complex document forms, affect the TL-BR alignment based on bounding box coordinates.

The SPADE decoder (Hong et al., 2022) operates robustly even with the incorrect order information by using the Initial Token Classification (ITC) and Subsequent Token Classification (STC) layer of the SPADE decoder. These two types of layers connect with LiLT, receiving the last hidden states output from LiLT to perform the downstream task. The ITC layer classifies the entity type for the initial token within the bounding box and the STC layer classifies which tokens are connected to each other for all tokens. In this process, it learns how tokens within the same semantic entity are connected in order. Therefore, to be form-independent, we used the SPADE decoder in our pipeline.

4 Experiment Setting

To assess our pipeline's performance, particularly the model, we conduct experiments on two types of Low-Resource Language (LRL) business documents and two open datasets as shown in Table 1. Due to personal information security concerns, these datasets are not publicly available.

Dataset	Language	Type	# of Entity	Train	Valid	Test
Medical bills	Ko	Forms	68	829	98	-
Receipts	Ja	Receipts	16	990	110	-
FUNSD	En	Forms	3	149	50	-
CORD	En	Receipts	30	800	100	100

Table 1: Information on LRL business documents and open datasets that were used to train for SER. # of Entity refers to the total number of unique entities.

4.1 LRL business documents

Korean medical bills contain diverse medical and financial information from various Korean hospitals, including detailed patient records, treatment specifics, complex pricing tables, and hospital details. They come in various formats such as faxes, scans, and mobile phones. Japanese receipts are general Japanese receipts similar to CORD (Park et al., 2019) in Japanese. These documents contain information about the store name, expenditure details, taxes, etc, also in various types including mobile photos.

Data preprocessing: We utilize the LOFI pipeline described in Section 3. Our validation dataset includes both clean images and manually selected examples with rotation, distortion, and low resolution, reflecting real-world conditions to assess the pipeline's robustness in diverse practical implementation settings.

Model setting: For our SER experiments, we employ various PLMs as text encoders, as we named LOFI-en, LOFI-ko, LOFI-ja, LOFI-mul†, LOFI-mul‡, and LayoutXLM^{o} (SCUT-DLVCLab, 2024; KLUE, 2024; Ku-NLP, 2024; Facebook AI, 2024; Microsoft, 2024). As you can see from Table 2, All models starting with LOFI- are based on the LiLT model combined with a SPADE decoder. Consistently across all configurations, we

Name	Language	Encoder	Parameters	Modality	Image Embedding	Korean medical bills	Japanese receipts
LayoutXLM°	Multi	$\text{LayoutXLM}_{\text{BASE}}$	369 M	T + L + I	ResNeXt101-FPN	95.58%	94.35%
LOFI-mul†	Multi	$\text{InfoXLM}_{\text{BASE}}$ + lilt-only-base	284 M	T + L	None	93.81&	**94.60%**
LOFI-mul‡	Multi	$\text{XLMRoBERTa}_{\text{BASE}}$ + lilt-only-base	284 M	T + L	None	94.24&	94.10%
LOFI-ko	Ko	$\text{RoBERTa}_{\text{BASE}}$ + lilt-only-base	116 M	T + L	None	**95.64%**	-
LOFI-ja	Ja	$\text{RoBERTa}_{\text{BASE}}$ + lilt-only-base	106 M	T + L	None	-	93.78%

Table 2: Entity-level F1 scores of the LRL business documents."T/L/I" denotes "Text/Layout/Image" modality.

use LiLT's layout encoder (lilt-only-base) (SCUT-DLVCLab, 2024) as the layout encoder layer. To compare with other methodologies that can process Korean or Japanese, our baseline model consisted of LayoutXLM combined with the initialized SPADE decoder weights.

4.2 Open datasets

We used FUNSD (Jaume et al., 2019) and CORD (Park et al., 2019) to see the performance on English datasets.

Data preprocessing: We use standardized preprocessing for fair model comparison: 1) Use original dataset text and bounding boxes. 2) Construct 1D input sequence using dataset-provided order. 3) Use dataset word-level bounding boxes without token-level splitting. See Table 1 for dataset details.

Model setting: For English datasets (FUNSD & CORD), we combine LiLT-RoBERTa-en-base (SCUT-DLVCLab, 2024) with the SPADE decoder, denoted as LOFI-en. LayoutLM, LayoutLMv2, LayoutLMv3, LiLT, BROS use BIO tagging for SER.

5 Experiment Results

We use the entity-level F1 score as the measure standard (Wei et al., 2020) for both experiments.

5.1 LRL business documents

Table 2 presents the entity-level F1 score for LRL business documents. For Korean medical bills, LOFI-ko demonstrated relatively higher performance on Korean documents, a LRL target, without additional pre-training or vision information, when compared to LayoutXLM. Furthermore, with only 116M parameters, approximately 68.6% fewer than LayoutXLM, our model offers significant advantages in resource utilization and processing speed.

For Japanese receipts, the multilingual model combining lilt-infoxlm-base with a SPADE decoder demonstrated the relatively higher performance, surpassing LayoutXLM while using fewer parameters and computational resources.

These findings highlight the effectiveness of our approach for Korean and Japanese documents, even in the absence of specific PLMs. F1 scores across various language models indicate broad applicability to diverse languages and document types. Interchangeable text encoders allow adaptation to industry needs. Our results demonstrate the model's effectiveness and potential for practical applications, especially where resource constraints and multilingual capabilities are crucial.

5.2 Open datasets

Name	Parameters	Modality	Image Embedding	FUNSD	CORD
LayoutLM	160 M	T + L	ResNet-101 (fine-tune)	79.27 %	94.72 %
LayoutLMv2	200 M	T + L + I	ResNeXt101-FPN	82.76 %	94.95 %
LayoutLMv3	133 M	T + L + I	Linear	79.38 %	**96.80 %**
BROS	110 M	T + L	None	**83.05 %**	95.73 %
LOFI-en	131 M	T + L	None	78.99 %	96.39 %

Table 3: Entity-level F1 scores of FUNSD and CORD datasets.

Table 3 shows the F1 scores for FUNSD (Jaume et al., 2019) and CORD (Park et al., 2019). For LayoutLMv3, we used word-level bounding boxes for direct comparison. LOFI-en also used word-level boxes without token-level splitting. BROS led FUNSD (83.05%), while LayoutLMv3 led CORD (96.80%).

LOFI-en was similar to LayoutLMv3 on CORD (96.39%) but trailed BROS by 4% on FUNSD (78.99%). This reveals LOFI's need for ample fine-tuning data, evident in performance differences between CORD (800 documents) and FUNSD (149 documents). The results highlight LOFI's limitations with limited fine-tuning data compared to pre-trained models.

6 Ablation Study

6.1 Number of training data for fine-tuning

Additionally, we conducted an experiment to determine the minimum number of training samples needed for satisfactory SER fine-tuning performance. The experiment compared performance across training data sizes ranging from 50 to 400 documents for Korean medical bills and Japanese receipts using LOFI-ko, LOFI-ja, LOFI-mul†, and LOFI-mul‡ models. Figure 2 demonstrates how

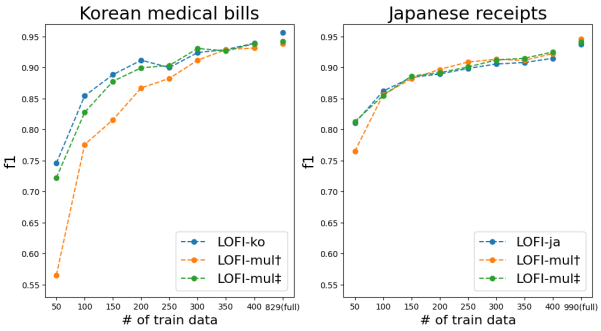


Figure 2: Performance change based on the number of fine-tuning training data samples. The x-axis represents the number of train data. The y-axis represents the entity-level F1 score.

performance varies with different number of training data in SER.

While the required number of training data may differ based on language, document structure, and characteristics, achieving satisfactory performance typically requires at least 300-400 documents. With fewer than 200 training documents, there is at least 5% performance difference compared to using the full training dataset. Given the time and cost constraints of building a large training dataset, research into methods for achieving robust performance with fewer training data is crucial.

6.2 Layout encoder layers

Layout encoder	Korean medical bills	Japanese receipts
Pre-trained	0.9564	0.9290
Initialized	0.9259	0.9035

Table 4: Comparison of entity level f1 score based on the use of pre-trained layout encoder weight. In random initialization, the weights are drawn from a zero-mean Gaussian distribution.

We tested LRL business documents to see language's impact on layout encoder, as shown in Table 4. Using Korean & Japanese RoBERTa for text encoding, we compared performance with and without English-based weights (lilt-only-base) for the layout encoder layers. The LOFI pipeline, employing pre-trained layout layers weights, showed 3.05% higher performance on detailed statements and 2.55% higher on Japanese receipts.

7 Use Cases

7.1 Automation of claim document processing for Korean insurance companies

Korean insurance companies have recently launched remote claim services, allowing customers to submit documents via phone, fax, or scanned emails. This surge in remote claims has increased the document processing workload. To overcome this issue, insurance companies have begun adopting document processing service. Figure 3 represents an example image of a Korean detailed medical bill used to process in LOFI pipeline.

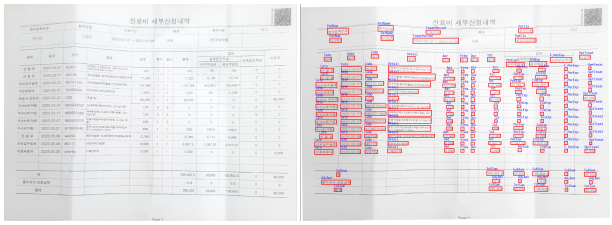


Figure 3: Before and after a Korean detailed medical bill image goes through LOFI pipeline

Our LOFI pipeline addresses specific needs in this industry: protecting customer privacy by processing documents on-premises, handling visual noises in document images such as blur and distortion of images from various channels, and managing numerous document types with format variations across institutions. This requires scalability and efficiency within limited computing resources.

The LOFI pipeline successfully automated various insurance claim documents process. Clients verified that our pipeline achieved an average accuracy of 97% across different document types. This resulted in a reduction of processing time by over 60% and a decrease in staff requirements by 40%. This case demonstrates the LOFI pipeline's effectiveness in addressing complex document processing challenges in the Korean insurance industry.

7.2 Automation of receipts processing for Japanese application service company

A Japanese application service company developed a tax-free declaration service to assist small retail shops. Retailers can now register passport photos and receipt information through a smartphone app. The service company then compares the entered receipt content with the captured receipt image and handles the tax agency declaration on behalf of the retailer. Initially relying on manual data entry, the growing service required automation. Our LOFI pipeline was implemented to automate receipt processing, addressing challenges posed by Japanese text characteristics and varying receipt layouts. The lack of spacing in Japanese text on receipts poses challenges for DIE.

Therefore, through collaboration, we applied the LOFI pipeline to the tax-free declaration service,

developing an automated function for the product information input and verification process. This demonstrates the LOFI pipeline's effectiveness in handling complex document processing tasks in LRL and complex document formats.

8 Conclusions and Future Work

In this paper, we propose LOFI, a DIE pipeline for SER tasks in Low-Resource Language (LRL) business documents. The LOFI pipeline extracts text and bounding boxes from image documents via OCR, preprocesses them using a token-level box split logic, and performs SER fine-tuning without pre-training by replacing the PLM. It achieves language independence through PLM replacement, OCR independence via token-level box split logic, and form independence by extracting information despite image rotation or distortion. Demonstrated on Korean and Japanese datasets, we anticipate its applicability to other LRL business documents.

Future research will focus on data augmentation, efficient annotation, and improved decoder architectures to handle document challenges to enhance AI capabilities for diverse business scenarios and document types.

9 Limitations

The practical implementation of the LOFI pipeline in the industry is constrained by the need for extensive training data. For instance, insurance companies dealing with Korean medical policies must process a wide variety of medical documents, each requiring specialized knowledge for accurate annotation, and Korean medical bills is one of them. The creation of training datasets is restricted by the need for domain expertise, time-intensive labor, and the complexity of establishing clear annotation guidelines. Also, the documents used in the experiment cannot be reproduced because they contain security policies and sensitive personal information.

Moreover, the LOFI pipeline's encoder-based model is susceptible to OCR errors deriving from low-quality images or noise, as it relies directly on OCR output for information extraction. For real-world automation, addressing these limitations is crucial. Future research will focus on developing methods to decrease the impact of OCR errors and post-processing the results, thereby enhancing the robustness and applicability of document information extraction systems in diverse business contexts.

10 Ethics Statement

Our research focuses on developing a language, OCR, and form independent pipeline to enhance DIE efficiency in industrial applications. Throughout this process, we adhered strictly to ethical guidelines, including those set by the EMNLP conference for data usage. As researchers, we take full responsibility for the study's ethical integrity and are committed to maintaining the highest standards in DIE research. This approach reflects our understanding of the broader implications of our work, balancing technological advancement with ethical considerations to ensure our contributions are both innovative and responsible.

Acknowledgments

References

Tom Bryan, Jacob Carlson, Abhishek Arora, and Melissa Dell. 2023. Efficientocr: An extensible, open-source package for efficiently digitizing world knowledge. *arXiv preprint arXiv:2310.10050*.

Yufan Chen, Jiaming Zhang, Kunyu Peng, Junwei Zheng, Ruiping Liu, Philip Torr, and Rainer Stiefelhagen. 2024. Rodla: Benchmarking the robustness of document layout analysis models. *arXiv preprint arXiv:2403.14442*.

Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. 2021. Document ai: Benchmarks, models and applications. *arXiv preprint arXiv:2111.08609*.

Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. 2023. Vision grid transformer for document layout analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19462–19472.

Facebook AI. 2024. Xlm-roberta-base. https://huggingface.co/FacebookAI/xlm-roberta-base.

Shohei Higashiyama, Masao Ideuchi, Masao Utiyama, Yoshiaki Oida, and Eiichiro Sumita. 2022. A japanese corpus of many specialized domains for word segmentation and part-of-speech tagging. In *Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems*, pages 1–10.

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.

Geonuk Kim, Jaemin Son, Kanghyu Lee, and Jaesik Min. 2022. Character decomposition to resolve class imbalance problem in hangul ocr. *arXiv preprint arXiv:2208.06079*.

Nikolaj Kjøller Bjerregaard, Veronika Cheplygina, and Stefan Heinrich. 2022. Detection of furigana text in images. *arXiv e-prints*, pages arXiv–2207.

KLUE. 2024. Roberta-base (korean). https://huggingface.co/klue/roberta-base.

Ku-NLP. 2024. Roberta-base japanese char wwm. https://huggingface.co/ku-nlp/roberta-base-japanese-char-wwm.

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.

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.

Microsoft. 2024. Infoxlm-base. https://huggingface.co/microsoft/infoxlm-base.

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*.

SCUT-DLVCLab. 2024. Lilt-roberta-en-base. https://huggingface.co/SCUT-DLVCLab/lilt-roberta-en-base.

Jaehyung Seo, Hyeonseok Moon, Jaewook Lee, Sugyeong Eo, Chanjun Park, and Heui-Seok Lim. 2023. Chef in the language kitchen: A generative data augmentation leveraging korean morpheme ingredients. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6014–6029.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020. Improving chinese word segmentation with wordhood memory networks. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8274–8285.

Jiapeng Wang, Lianwen Jin, and Kai Ding. 2022. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. *arXiv preprint arXiv:2202.13669*.

Jiapeng Wang, Chongyu Liu, Lianwen Jin, Guozhi Tang, Jiaxin Zhang, Shuaitao Zhang, Qianying Wang, Yaqiang Wu, and Mingxiang Cai. 2021. Towards robust visual information extraction in real world: new dataset and novel solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2738–2745.

Renshen Wang, Yasuhisa Fujii, and Alessandro Bissacco. 2023. Text reading order in uncontrolled conditions by sparse graph segmentation. In *International Conference on Document Analysis and Recognition*, pages 3–21. 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*, 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.

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*.

Chong Zhang, Ya Guo, Yi Tu, Huan Chen, Jinyang Tang, Huijia Zhu, Qi Zhang, and Tao Gui. 2023. Reading order matters: Information extraction from visually-rich documents by token path prediction. *arXiv preprint arXiv:2310.11016*.

A Appendix

A.1 Fine-tuning configuration

Dataset	Train Epoch	Learning Rate	Batch Size	Max Length
Korean medical bills	50	1e-5	24	512
Japanese receipts	100	5e-5	32	512
FUNSD	100	5e-5	4	512
CORD	100	5e-5	16	512

Table 5: Hyperparameter setting for LRL business documents and open datasets.

The base configurations for all models in our experiments are 768 hidden size, 12 self-attention heads, 3072 feed-forward size, and 12 encoder layers. This standardized approach to model architecture and fine-tuning allows for more meaningful comparisons across different language models and datasets.

A.2 Text alignment algorithm

Algorithm 1 Top-Left to Bottom-Right text alignment algorithm

Require: Set of bounding boxes B, height tolerance ϵ
Ensure: Sorted list of bounding boxes S

function SORTBOUNDINGBOXES(B, ϵ)
  $S \leftarrow \emptyset$
  while $B \neq \emptyset$ **do**
    $R \leftarrow \emptyset$ ▷ Current row
    $h_{ref} \leftarrow$ HEIGHT($B[1]$)
    for $box \in B$ **do**
      if $|\text{HEIGHT}(box) - h_{ref}| \leq \epsilon$ **then**
        $R \leftarrow R \cup \{box\}$
      end if
    end for
    Sort R from left to right
    $S \leftarrow S \cup R$
    $B \leftarrow B \setminus R$
  end while
  return S
end function
function HEIGHT(box)
  return $box.height$
end function

procedure MAIN
  $B \leftarrow$ LOADBOUNDINGBOXES('path')
  $\epsilon \leftarrow$ predefined tolerance value
  $S \leftarrow$ SORTBOUNDINGBOXES(B, ϵ)
  Output S
end procedure

Algorithm 1 is designed to sort all bounding boxes extracted by OCR engine. It compares the differences in y-axis positions between boxes. If the absolute difference is below a certain threshold, the boxes are considered to be on the same line. Starting from the box with the smallest y-coordinate value, it sequentially identifies and stores boxes that are on the same line. By repeating this process for all boxes, we obtain a sorted result that utilizes the layout of the bounding boxes.

A.3 Word-level and segment-level bounding boxes

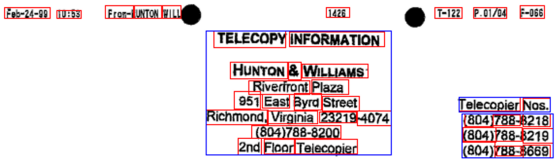


Figure 4: Blue boxes represent segment-level bounding boxes and red boxes represent word-level bounding boxes

Figure 4 illustrates an example visualization of layout information from the FUNSD dataset, showing both segment-level bounding boxes and word-level bounding boxes. Segment-level bounding boxes represent the layout information for the entire range of important information, known as entities. Word-level bounding boxes provide layout information at the individual word level. As evident from the figure, segment-level bounding boxes, which represents entity layouts, can encompass multiple word-level bounding boxes.

A.4 Token-level box split

Algorithm 2 shows the logic for converting text and bounding boxes extracted by OCR engine into tokens and token boxes. We use the model's tokenizer to tokenize the text. The resulting tokens are then divided into character units to determine the text type, which refers to character-level classifications such as numbers, special symbols, uppercase letters, and lowercase letters. This classification is necessary because the character size in documents vary by type. Then we pre-define the ratios that exist for each character type. By using these ratios to split the bounding box proportionally for each token, we determine token boxes that correspond to the size of each token. This process is applied uniformly to all text and bounding boxes, which then we are able to obtain a result where the original inputs are split into tokens and corresponding

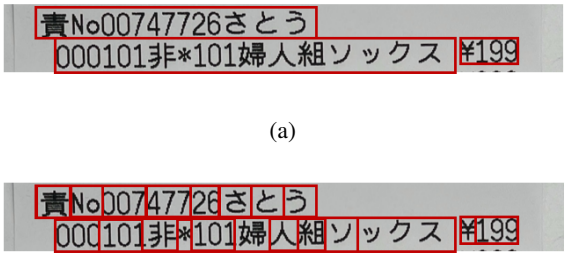
token boxes.

Algorithm 2 Token-Level box split Algorithm

Require: Image I, OCR engine O, Tokenizer T
Ensure: Tokenized text T, Bounding boxes B

function TOKENLEVELBOXSPLIT(I, O, T)
  $(text, boxes) \leftarrow O(I)$ ▷ Perform OCR
  $T \leftarrow T(text)$ ▷ Tokenize text
  $C \leftarrow$ IDENTIFYCHARTYPES(T)
  $B \leftarrow$ CALCTOKENBOXES($T, C, boxes$)
  return T, B
end function
function IDENTIFYCHARTYPES(T)
  $C \leftarrow \{\}$
  for each $token$ in T **do**
    $c_{token} \leftarrow$ [GETCHARTYPE($char$) for each $char$ in $token$]
    $C \leftarrow C \cup \{c_{token}\}$
  end for
  return C
end function
function CALCTOKENBOXES($T, C, boxes$)
  $B \leftarrow \{\}$
  for $i \leftarrow 1$ to $|T|$ **do**
    $size_i \leftarrow \sum_{j=1}^{|C_i|}$ GETBOXSIZE($C_i[j]$)
    $B \leftarrow B \cup \{$ADJUSTBOX($boxes[i], size_i$)$\}$
  end for
  return B
end function
function GETCHARTYPE($char$)
  return CharacterClassification($char$) ▷ Returns character type classification
end function
function GETBOXSIZE($char\_type$)
  return PredefinedSizeRatio($char\_type$) ▷ Returns size ratio based on character type
end function
function ADJUSTBOX($box, size$)
  return ModifiedBox($box, size$) ▷ Adjusts original OCR box based on calculated size
end function

Figure 5 shows an example of before and after token-level box split algorithm is applied. Figure 5 (a) represents an example of text and bounding boxes extracted by our OCR engine from a Japanese receipt image. (b) illustrates the result after applying the algorithm. This shows more meaning-based bounding boxes to give more accurate results.



(a)

(b)

Figure 5: (a) represents the bounding boxes extracted from the OCR engine and (b) represents token unit boxes divided by token-level box split algorithm.

A.5 Annotation

We describe the annotation process for the training and evaluation data and our experience with it.

1. We first reviewed open datasets and sought to understand the business processes involving the documents. Through this separate process, we were able to construct an annotation guidance framework.

2. To minimize subjective judgements, we collected and discussed exceptional cases that arose during the annotation process and revised the annotation guidance accordingly. Multiple annotators could perform the task simultaneously using an annotation tool.

3. Finally, we ensured higher data quality by having different annotators cross-check each other's work, resulting in a cleaner and more reliable dataset.

During the annotation process, we also conducted model training with qualitative evaluations, confirming that the inference results improved through the process mentioned above.

Step	**Number of people / period**	
	Korean medical bills	**Japanese receipts**
Research	2 people / 2 weeks	1 person / 4 weeks
Annotation	4 people / 5 weeks	2 people / 2.5 weeks
Inspection	3 people / 4 weeks	2 people / 1 week

Table 6: Duration and personnel required for each annotation stage of Korean medical bills and Japanese receipts.

A.6 Supplementary Data Information

Dataset	**Type**	**Length**	**Total entities**
Korean medical bills	Train	1370	410,735
Korean medical bills	Valid	1233	21,337
Japanese receipts	Train	293	21,731
Japanese receipts	Valid	280	2,572
FUNSD	Train	845	7,411
FUNSD	Valid	1011	2,332
CORD	Train	118	11,106
CORD	Valid	103	1,247
CORD	Test	113	1,336

Table 7: Length refers to the average text length that appears on a single image. Total entities refers to the total number of entities across all images.

Table 7 shows the statistics of the datasets used in the experiment. Tables 8 and 9 show how we defined the entity classes for the Korean medical bills and Japanese receipt data.

As for the korean medical bills, we defined the entities in a format necessary for real-world scenarios as follows. For detailed estimation reports, entities are categorized into those outside the table and those inside the table. To differentiate between these, entities are composed of Key and Value. Key: A unique item that serves as a reference point for locating a specific value, and does not repeat. Value: The value corresponding to the Key. Entities inside the table are composed of Head and Line. Head: The equivalent of a column name in a table, and is a unique, non-repeating item. Line: The value corresponding to the Head, which may be a repeated item. We distinguished each entity as either Key, Value, Head, or Line depending on whether it is outside or inside the table. To provide a clearer understanding, we will share some examples of the entities defined in Korean medical bills.

As for the Japanese receipts, we only constituted of Key, Value. We share all the entities in the following table.

Type	Entity	Appearance	Description
Patient	ID	Key/Value	A unique identifier assigned to the patient within the hospital's system.
	Name	Key/Value	The full legal name of the patient.
	Period	Key/Value	The timeframe during which the medical history statement is relevant. This could specify the duration of the patient's treatment, admission dates, or the period over which the medical services were provided.
	Class	Key/Value	This may refer to the classification of the patient's insurance, the category of service (e.g., inpatient, outpatient), or another relevant classification system used by the hospital to categorize patients.
Hospital	Name	Key/Value	The official name of the hospital or medical facility.
	Representative	Key/Value	The name or title of the hospital representative responsible for the medical history statement.
	Subject	Key/Value	The main topic or purpose of the medical history statement.
Medical Treatment	Category	Head/Line	The classification of the medical treatment or service.
	Date	Head/Line	The date when the medical treatment or service was administered.
	Item	Head/Line	A description of the specific medical service, procedure, medication, or item provided to the patient.
	Item Code	Head/Line	A standardized code associated with the medical item or service.
	Number of Days	Head/Line	The duration for which a particular treatment or service was administered, measured in days.
	Quantity/Dose	Head/Line	The amount of medication administered or the quantity of a service provided.
	Unit Price	Head/Line	The cost per single unit of the medical item or service.
	Price	Head/Line	The total cost for the specific medical item or service, typically calculated as Quantity/Dose multiplied by Unit Price.
Total	Total	Key/Value	The aggregate amount due for all medical treatments and services listed.
	Subtotal	Key/Value	The intermediate total calculated by summing amounts grouped by Category or Date.

Table 8: Descriptions of entity types and their corresponding keys and values in korean medical bills. Although the total number of unique entities is 68, only representative entities are shown here.

Type	Entity	Appearance	Description
Store	Name	Only Value	The name of the store or seller.
Product	Name	Only Value	The name of the product or item.
	Code	Only Value	The code of the product or item.
	Quantity	Only Value	The quantity of the product purchased.
	Unit price	Only Value	The price per unit of the product.
	Price	Only Value	The total price for this product (quantity * unit price).
	Tax	Key/Value	Tax information for the product.
	Discount	Key/Value	The discount amount.
Payment	Subtotal	Key/Value	The total subtotal amount (before tax and discounts).
	Total	Key/Value	The final total amount (after tax and discounts).
	Tax total	Key/Value	The total tax amount.

Table 9: Descriptions of entity types and their corresponding keys and values in japanese receipts.