Lightweight Spatial Modeling for Combinatorial Information Extraction From Documents

Yanfei Dong[1,2], **Lambert Deng**[1], **Jiazheng Zhang**[1]
Xiaodong Yu[1], **Ting Lin**[1], **Francesco Gelli**[1], **Soujanya Poria**, **Wee Sun Lee**[2]
[1] PayPal [2] National University of Singapore
DeCLaRe Lab, Singapore University of Technology and Design
{dyanfei, yuadeng, jiazzhang, xiaodyu, tinlin, fgelli}@paypal.com
sporia@sutd.edu.sg, leews@comp.nus.edu.sg

Abstract

Documents that consist of diverse templates and exhibit complex spatial structures pose a challenge for document entity classification. We propose *KNN-Former*, which incorporates a new kind of spatial bias in attention calculation based on the K-nearest-neighbor (*KNN*) graph of document entities. We limit entities' attention only to their local radius defined by the *KNN* graph. We also use combinatorial matching to address the one-to-one mapping property that exists in many documents, where one field has only one corresponding entity. Moreover, our method is highly parameter-efficient compared to existing approaches in terms of the number of trainable parameters. Despite this, experiments across various datasets show our method outperforms baselines in most entity types. Many real-world documents exhibit combinatorial properties which can be leveraged as inductive biases to improve extraction accuracy, but existing datasets do not cover these documents. To facilitate future research into these types of documents, we release a new ID document dataset that covers diverse templates and languages. We also release enhanced annotations for an existing dataset.[1]

1 Introduction

Structured document information extraction (IE) attracts increasing research interest due to the surging demand for automatic document processing, with practical applications in receipt digitization, workflow automation, and identity verification etc.

Recent state-of-the-art methods for processing documents with complex layouts extensively exploit layout information, such as position, relative distance, and angle, with transformer-based models. Spatial modelling is a key contributing factor to the success of these methods (Xu et al. 2020, Appalaraju et al. 2021, Xu et al. 2021, Hwang et al. 2021). However, absolute coordinates, pair-wise relative Euclidean distance, and angle are insufficient to capture the spatial relationship in complex layouts. Two document entity pairs could carry different importance despite having the same position and distance, due to the presence or absence of other entities positioned between the pairs. We believe that spatial information can be better exploited for document entity classification.

We propose *KNN-Former*, a parameter-efficient transformer-based model that extracts information from structured documents with combinatorial properties. In addition to relative Euclidean distance and angle embeddings as inductive biases (Hwang et al., 2021), we introduce a new form of spatial inductive bias based on the K-Nearest Neighbour (*KNN*) graph which is constructed from the document entities and integrate it directly into the attention mechanism. Specifically, we first construct a *KNN* graph based on the relative Euclidean distance of document entities. Then we incorporate hop distance between entities, which is defined as the shortest path between two entities on the *KNN* graph, in training their pair-wise attention weight. For entity pairs with the same Euclidean distance but different hop distance, the difference in hop distance would still contribute to different attention weights. We limit an entity's attention calculation only to its local radius of neighborhood defined by the *KNN* graph. This also strengthens the inductive bias as reflected by our experiment results.

Furthermore, many real-world document information extraction tasks come with combinatorial properties, such as one-to-one mapping between field categories and values. Such combinatorial properties can be leveraged as inductive biases to improve the extraction accuracy, but are underexplored because existing datasets do not cover such documents. Current methods that do not address the combinatorial constraints suffer suboptimal performance on these types of documents. We further leverage this inductive bias by treating the

[1] https://github.com/miafei/knn-former

Findings of the Association for Computational Linguistics: EACL 2023, pages 1471–1484
May 2-6, 2023 ©2023 Association for Computational Linguistics

entity classification task as a set prediction problem and using combinatorial matching to post-process model predictions(Kuhn, 1955; Carion et al., 2020; Stewart et al., 2016).

In addition, *KNN-Former* is parameter-efficient. Recent baseline models are initialized with parameters of pre-trained language models (Xu et al., 2020, 2021; Hwang et al., 2021; Hong et al., 2022), making their model size larger or at least comparable to the language models. *KNN-Former* does not utilize initialized parameters of existing language models, therefore free from the parameter size floor restriction. It is designed to be 100x smaller in trainable parameters compared to prevailing baselines. *KNN-Former*'s parameter efficiency makes it energy-efficient, contributes to faster training, fine-tuning and inference speed and makes mobile deployment feasible.

To encourage the progress of IE research in complex structured documents with combinatorial mapping properties, we release an ID document dataset (named POI). While the existing ID document dataset has only 10 templates (Bulatov et al., 2021), POI exhibits better template and lingual diversity. It also has a special mapping constraint where one field category has only one corresponding entity. In compliance with privacy regulations, the documents in the POI dataset are specimens and do not contain information about real persons.

We conduct extensive experiments to evaluate the effectiveness of our proposed method. *KNN-Former* outperforms baselines on most field categories across various datasets, despite having a significantly smaller model size. Extensive ablation studies show the importance of the *KNN*-based inductive bias and combinatorial matching.

To summarize, our contributions include (1) a highly parameter-efficient transformer-based model that (2) incorporates *KNN*-based graph information in ~~sparsified~~ local attention; (3) combinatorial matching to address the one-to-one mapping constraint; (4) a new ID document dataset with good template diversity, complex layout, and a combinatorial mapping constraint.

2 Related Work

Researchers have tried multiple approaches for document information extraction (Jaume et al., 2019; Mathew et al., 2021; Stanisławek et al., 2021). However, these works do not have spatial cues, such as the position of the information in the original document. To address this shortcoming, a number of works introduce the modality of layout information as additional input features. Majumder et al. (2020) adopts positional information as inputs to their method to extract information from receipt documents. LayoutLM (Xu et al., 2020) adds 1-D and 2-D absolute position encodings to text embeddings before passing them to the transformer. Hong et al. (2021) proposes to train a language model from unlabeled documents with area masking, encoding relative positions of texts. StructuralLM (Li et al., 2021) assigns the bounding box cell position as the position coordinates for each word contained in it. DocFormer (Appalaraju et al., 2021) encodes 2D spatial coordinates of bounding boxes for visual and language features. LayoutLMv2 (Xu et al., 2021) uses learnable pair-wise relative positional embeddings as attention bias.

A few works propose to use graphs to represent spatial entity relationships in documents. SPADE (Hwang et al., 2021) uses a three steps graph decoder and formulates the information extraction task as a dependency parsing problem. FormNet (Lee et al., 2022) constructs a k-nearest neighbor graph and applies a 12-layer graph convolutional network (GCN) to get the entity embeddings before feeding them into a transformer network. However, there are some limitations in using GCN to obtain embeddings. It is well established that the message passing-based GCN are limited in their expressive power (Xu et al., 2018; Arvind et al., 2020; Morris et al., 2019; Chen et al., 2020; Loukas, 2019; Dehmamy et al., 2019). In addition, FormNet does not use the hop distance between nodes, which could serve as a strong inductive bias to capture the spatial relationships between document entities.

Datasets with positional information such as Funsd (Jaume et al., 2019), Cord (Park et al., 2019), Sroie (Huang et al., 2019) are released to facilitate research in document understanding. However, they do not contain documents with combinatorial properties which are common in real-world applications.MIDV500 (Arlazarov et al., 2018) and MIDV2020 (Bulatov et al., 2021) are two synthetic ID datasets with combinatorial properties, but are unsuitable for document information extraction tasks due to incomplete annotations. They also lack template diversity.

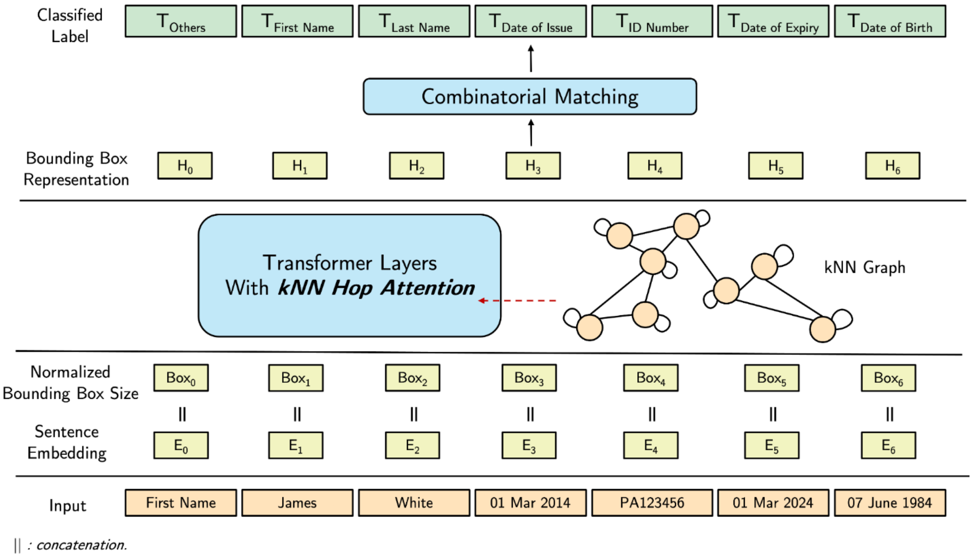


Figure 1: An illustration of *KNN-Former*. Bounding box texts are embedded using sentence transformer, which are concatenated with embeddings of bounding box size to form input embeddings. The concatenated embeddings are then passed to the transformer layers with *KNN* Hop Attention, which incorporates pair-wise relative hop distance between entities on *KNN* graph in attention calculation. The output entity representations of the transformer layers are passed to combinatorial matching for set prediction.

3 Methodology

In this section, we discuss the methodology for our model. We formulate the problem in Sec.3.1 and explain our overall model architecture and the details of each component in Sec.3.2.

3.1 Problem Formulation

Given a document D which consists of multiple entities $\{e_i, \ldots, e_j\}$, and the bounding box coordinates and texts$\{x_i, \ldots, x_j\}$ detected by Optical Character Recognition (OCR) tool. We measure the relative distance and angle between two entities e_i and e_j as $\sigma_{(i,j)}$ based on the coordinates of bounding boxes. Our task is to map each entity e_i in document D to its field category y_i, which is one of the predefined labels. For each field category y_i, there is only one corresponding entity e_i.

3.2 Model Architecture

We propose *KNN-Former*, a transformer-based model for document entity classification. The architecture of *KNN-Former* is shown in Fig. 1. *KNN-Former* uses K-Nearest Neighbours Hop Attention, which incorporates a new inductive bias into attention computation. *KNN-Former* also treats document entity classification as a set prediction problem and uses combinatorial assignment to address the one-to-one correspondence between entities and fields. *KNN-Former* is highly parameter-efficient compared to baselines. Details of model size can be found in Tab 4.

3.2.1 K-Nearest Neighbors Hop Attention

One key contribution of *KNN-Former* is the proposed attention mechanism. Following (Lee et al., 2022), we first construct a *KNN* graph based on the Euclidean distance between each pair of entities. We represent entities as nodes and then connect edges between each entity and its K nearest neighboring entities. We also add a self-loop to each entity to improve performance (Kipf and Welling, 2016). While previous works focus on leveraging pair-wise relative Euclidean distance (Xu et al., 2021; Hwang et al., 2021), we propose to incorporate pair-wise relative **hop distance**, which is defined as the shortest path between two entities on the *KNN* graph. Two entities could be in proximity in terms of Euclidean distance but not so in terms of hop distance. For example, in documents with complex layouts, it is common to have two entities that are close to each other in the Euclidean space, but there is a third entity positioned in between. This type of entity pair should be treated differently from pairs that are close to each other in both Euclidean and hop distances. In this case, the spatial attention mechanism based solely on the relative Euclidean distances between entity pairs is insufficient since it neglects this structural information. We argue that the *KNN* graph structure is an effective way of capturing the structural information and propose to incorporate it as an inductive bias into the attention computation.

Intuitively, different hop distances should carry different weights in calculating pairwise attention. We use $\phi_{(i,j)}$ to represent the hop distance between entity i and j and H to represent a learnable embedding lookup table based on the hop distance $\phi_{(i,j)}$. Inspired by DeBERTa (He et al., 2020) and Transformer-XL (Dai et al., 2019), we integrate the hop distance bias into attention as described in the following equations

$$\begin{aligned} e_{ij} = [x_i W^Q (x_j W^K + H^Q_{\phi_{(i,j)}} + R^Q_{\sigma_{(i,j)}}) \\ + (H^K_{\phi_{(i,j)}} + R^K_{\sigma_{(i,j)}}) x_i W^K] / \sqrt{d}, \end{aligned} \quad (1)$$

$$z_i = \sum_j a_{ij} (x_j W^V + H^V_{\phi_{(i,j)}} + R^V_{\sigma_{(i,j)}}), \quad (2)$$

where $\sigma_{(i,j)}$ is a concatenation of the relative Euclidean distance and angle between entity i and j, and R is a learnable matrix. H could be a learnable matrix or a lookup table that maps $\sigma_{(i,j)}$ to learnable embeddings. e_{ij} is the attention weight between entity i and j. a_{ij} is calculated as the weight of $exp(e_{ij})$ in the exponential sum of all e_{ik}, as described in Eqn.3.

$$a_{ij} = \frac{exp(e_{ij})}{\sum_k exp(e_{ik})}. \quad (3)$$

Similar to how pair-wise relative Euclidean distance is added to attention, we add pair-wise hop distance as three learnable weight matrices, two of which multiply with query and key vectors respectively while the remaining one is added to the value vector. We further limit an entity's attention only to its local radius of neighborhood defined by the *KNN* graph. Specifically, we do not calculate e_{ij} if the hop distance between entity i and j exceeds a certain threshold. This also strengthens the inductive bias as supported by our experiment results.

3.2.2 Combinatorial Matching

We hypothesize that combinatorial properties between field categories and entities can be leveraged as inductive biases to improve extraction performance. Different from existing methods that treat the classification of each entity independently (Xu et al., 2021; Hwang et al., 2021; Lee et al., 2022), we propose to treat the entity classification task as a set prediction problem to exploit the one-to-one mapping constraint, where one field has one and only one corresponding entity. The combinatorial assignment is described in Eqn.4.

$$\tau_{opt} = argmin_{\tau} \sum_i^N L_{match}(y_i^{label}, y_{\tau_{(i)}}^{pred}), \quad (4)$$

where τ is an assignment, and L_{match} is the matching cost. N is the number of entities in a document. In practice, N is often much larger than the number of entities of interest. Therefore, we pad the number of ground truths to N in order to perform a one-to-one combinatorial assignment. This can be done with the Hungarian algorithm in polynomial time (Kuhn, 1955; Carion et al., 2020; Stewart et al., 2016).

4 Datasets

Many real-world documents exhibit combinatorial properties, such as a one-to-one mapping between between its fields and entities. However, existing public datasets do not cover documents with such properties (Jaume et al., 2019; Park et al., 2019; Huang et al., 2019). To fill the gap, we release a new ID document dataset POI, and enhanced annotations of MIDV2020. We also verify our method on a private dataset PRV. All 3 datasets exhibit combinatorial properties.

In addition, we design the POI dataset to be template-rich with diverse languages. We also design the enhanced MIDV2020 with a difficult split such that templates in testing are unseen during training. BERT alone without spatial information can achieve above 90% F1 on some existing datasets (Hong et al., 2022; Park et al., 2019; Huang et al., 2019), indicating relative sufficiency of leveraging text information alone. Yet in many real-world use cases, using text alone is insufficient. This motivates us to work on more challenging datasets where the exploitation of spatial information is important. Dataset statistics are summarized in 1 and Tab. 2. More details are as follows.

	#Train Doc.	**#Test Doc.**
POI	421	109
MIDV2020	500	200
PRV	3480	807

Table 1: Number of documents in training and testing.

Dataset	Avg # of Ent. per Doc.	Total # of Ent.	Total # of Doc.
POI	31.79	16850	530
MIDV2020	32.85	23000	700
PRV	24.31	104245	4287

Table 2: Statistics of entity distribution in documents. Ent. stands for entities and Doc. stands for documents.

POI We collect and annotate 530 Proof-of-Identity documents from online sources. We will release this POI dataset which consists of 10 document types, 265 distinct templates, and 131 countries of origin. The template and language diversity of POI create a challenging task for document understanding. All images are specimens with dummy values. The document type distribution is shown in Tab.3.

There are 8 field categories in total: last name, first name, date of birth, date of issue, date of expiry, ID number, key, and others. Key represents entities that indicate the field names for the important entities (e.g. Last Name) that we are interested to extract. The first 6 field categories appear in each document image once and only once, creating a special mapping constraint unseen in other datasets. The last 2 field categories (key and others) are not subject to the constraint. In real-world applications, it is common to extract a set of entities from documents that have combinatorial properties between its field and entities. ID document information extraction is one such use case, where we only expect to extract one entity for each field category of interest. This one-to-one correspondence can be leveraged to improve classification performance. Despite being a common task setting, we notice the lack of method exploration and innovation in this direction, due to the unavailability of such property among existing popular document datasets. More details about the dataset can be found in the Appendix.

Document Type	# Document
Passport	238
Driving License	119
Travel Document	109
ID	30
Resident Permit	21
Seafarer ID	10
Others	3

Table 3: Distribution of document types in POI dataset

MIDV2020 We utilize the 1000 synthesized ID documents from the initial MIDV2020 dataset (Bulatov et al., 2021) . These documents are generated from 10 templates, with 100 documents for each template. Each document image is annotated with a list of bounding box coordinates and field values. We find that only artificially generated entities, such as the values of names and ID numbers, are annotated, while entities that belong to the original templates, such as document title and field names are not. We proceed to annotate the remaining entities. The newly annotated ground truths of MIDV2020 will be released alongside POI. These enhanced annotations enable us to perform information extraction task in a setting that is closer to real-world application, where all texts recognized by the OCR engine are used. The train/test split we introduce for MIDV2020 is a split by countries, this ensures that the document templates in the training dataset are unseen in the testing dataset. The country split simulates real-world scenarios where the model extension to new countries or new versions of documents is needed. More details can be found in the Appendix.

PRV Since POI and MIDV2020 only contain specimens or artificially generated images, we run our model on a private dataset (named PRV) that mostly consists of US driver licenses. The documents are protected by strict privacy requirements and massive human annotations are not available as raw images are inaccessible. Therefore, we build automatic fuzzy labeling to annotate the ground truth.

Comparison on Datasets POI exhibits better template and language diversity. POI contains 265 templates from 131 countries, while MIDV2020 has 10 templates from 10 countries. The number of templates in PRV is unknown due to privacy-related limitations. In addition, POI consists of templates in a multitude of languages, whereas MIDV2020 and PRV dataset lack such diversity. Texts in POI and MIDV2020 are made up largely by artificial text which is more readable and clearer, while PRV contains real texts. POI and PRV samples are split randomly. Since MIDV2020 has only 10 templates, we split the samples by country to make the task more challenging. PRV is the easiest dataset among the three due to its lingual monotony and random split.

Dataset	Method	F1 Score						Input Modality	#Parameters	
		L.Name	F.Name	DoB	DoI	DoE	ID No.		Trainable	Total
POI	BERT$_{\text{BASE}}$	67.90	72.73	92.11	70.78	69.06	78.70	text	110 M	110 M
	GCN	45.35	56.08	85.62	62.37	62.32	70.65	text + layout	31.5 K	22.7M
	LayoutLM$_{\text{BASE}}$	87.03	86.88	93.93	86.23	87.72	83.12	text + layout	110 M	110 M
	LayoutLMv2$_{\text{BASE}}$	**90.58**	**89.26**	96.00	94.22	92.59	88.16	text + layout + image	199 M	199 M
	SPADE	73.73	78.63	90.09	89.59	90.27	83.98	text + layout	128 M	128 M
	BROS$_{\text{BASE}}$	82.39	82.76	94.16	91.41	88.32	83.18	text + layout	109 M	109 M
	KNN-former	83.57	82.18	**98.37**	**95.89**	**94.48**	**90.06**	text + layout	0.5 M	23.2M
MIDV2020	BERT$_{\text{BASE}}$	40.61	52.89	**100.00**	85.29	80.00	55.62	text	110 M	110 M
	GCN	32.03	43.09	99.50	99.00	79.76	43.82	text + layout	31.5 K	22.7M
	StructuralLM$_{\text{LARGE}}$	25.13	11.83	**100.00**	89.29	91.53	**99.50**	text + layout	355 M	355 M
	LayoutLM$_{\text{BASE}}$	47.65	15.10	**100.00**	97.96	80.16	67.97	text + layout	110 M	110 M
	LayoutLMv2$_{\text{BASE}}$	47.54	49.91	87.15	97.56	77.24	94.18	text + layout + image	199 M	199 M
	SPADE	48.91	45.54	79.90	63.47	60.85	60.34	text + layout	128 M	128 M
	BROS$_{\text{BASE}}$	23.31	23.78	98.50	70.83	18.27	85.39	text + layout	109 M	109 M
	KNN-former	**87.88**	**54.26**	**100.00**	**100.00**	**95.21**	69.65	text + layout	0.5 M	23.2M
PRV	BERT$_{\text{BASE}}$	71.32	76.39	97.72	88.78	86.22	87.21	text	110 M	110 M
	GCN	66.32	81.97	97.59	89.53	87.90	89.38	text + layout	31.5 K	22.7M
	StructuralLM$_{\text{LARGE}}$	93.72	93.27	**99.56**	98.86	99.21	97.86	text + layout	355 M	355 M
	LayoutLM$_{\text{BASE}}$	**95.36**	94.71	99.17	98.76	98.61	97.85	text + layout	110 M	110 M
	LayoutLMv2$_{\text{BASE}}$	95.26	95.31	99.52	99.29	99.36	**98.82**	text + layout + image	199 M	199 M
	SPADE	65.61	70.65	98.70	98.10	96.43	92.48	text + layout	128 M	128 M
	BROS$_{\text{BASE}}$	93.52	91.68	99.00	98.44	97.53	97.91	text + layout	109 M	109 M
	KNN-former	92.03	**96.81**	91.22	**99.68**	**99.47**	98.76	text + layout	0.5 M	23.2M

Table 4: Entity-level F1 score of *KNN-Former* compared to baselines. Column L.Name, F.Name, DoB, DoI, DoE and ID No. correspond to results of Last Name, First Name, Date of Birth, Date of Issue, Date of Expiry, and ID Numbers. GCN and *KNN-Former* have additional 22.7 M fixed parameters since we employed a light-weighted 6-layer sentence transformer (Reimers and Gurevych, 2019) to get the text embeddings.

5 Experiments

In this section, we conduct extensive experiments to evaluate our proposed *KNN-Former* on aforementioned datasets. We first compare our results with several baselines in Sec. 5.1. Then in Sec. 5.2, we evaluate the generalization ability of our method on unseen templates. We then conduct ablation studies in Sec.5.3 and Sec.5.4 to assess the effects of each component in *KNN-Former* and the impact of different K in the *KNN* graph.

5.1 Comparison with Baselines on Multiple Datasets

We first evaluate the performance of *KNN-Former* against multiple competitive methods. We choose base models for most of the baselines, because these are closest to *KNN-Former* in terms of the number of parameters. Brief description of baseline models as well as the implementation details of all the models can be found in Sec. A.1. We do not have results for StruturalLM on POI dataset because of an OOV error.

Tab.4 shows the entity-level classification performance. The results show that our method outperforms the baselines on most entity types across various datasets. In particular, *KNN-Former* outperforms LayoutLMv2$_{\text{BASE}}$, a state-of-the-art model that uses additional image features. We also observe that BERT performs poorly on these datasets, indicating the importance of exploiting spatial information.

Secondly, as shown in Trainable Param column in Tab.4, *KNN-Former* is highly parameter-efficient. All baselines except GCN have more than 100 million trainable parameters, while *KNN-Former* has only 0.5 million and is magnitudes smaller than competing methods. Even after adding the sentence transformer, *KNN-Former* has only 23.2 million parameters, still 5x smaller than baselines. The parameter efficiency has 4 benefits. First, it contributes to learning and inference time efficiency, with details illustrated in 5.5. Second, it allows for faster fine-tuning on new datasets and domains, especially in real-world use cases when training datasets are big and re-training requirements are frequent. Third, smaller model size and faster inference time make mobile deployment more feasible. Fourth, training, fine-tuning and inferring smaller models reduces power consumption and carbon footprint. Despite the smaller model size, *KNN-*

Former achieves comparable or better performance across datasets.

Thirdly, we observe that *KNN-Former* underperforms both LayoutLM$_{\text{BASE}}$ and LayoutLMv2$_{\text{BASE}}$ for name related entities in both POI and PRV datasets. The robustness of the two baselines in predicting names could be attributed to their extensive pre-training. The two baselines learn common names in pre-training, enabling them to predict names correctly regardless of context. However, despite no extensive pre-training, *KNN-Former* still outperforms BROS and StructuralLM which are also pre-trained on 11 million documents.

Fourthly, we observe all methods suffer performance degradation on MIDV2020, compared to the other two datasets. This is because in MIDV2020, training and testing documents are split by countries, templates in testing are not seen during training. In addition, MIDV2020 has only 6 templates in training data, which easily leads to overfitting. Detailed discussion on the generalization ability can be found in Sec. 5.2. we find that BERT outperforms several baselines with spatial modelling on names, this may be due to overfitting to limited number of training templates. We notice that our method do not perform well on id number entity. We conducted manual inspection on several error cases, and find that in many documents there exist two different types of id numbers(see Fig. 3(b)), but only one of them is labeled as id number according to the provided annotations. Our model sometimes predicts the other one as id number. This also explains the poor performance on id number for some other baselines.

Lastly, we notice that on the PRV dataset, *KNN-Former* performs poorly on DoB field, underperforming even GCN. *KNN-Former*'s performance on DoB drops after combinatorial matching, despite an overall increase macro average F1. This could be due to the presence of noise in groundtruth, since this dataset is annotated by automatic fuzzy labeling logic. Manual examination of a few documents confirms our hypothesis.

5.2 Evaluation of generalization ability on unseen templates

To assess the generalization capability of our model, we test and compare our model with other competitive baselines on MIDV2020 dataset using two train/test settings: random split and split by country . The country split is a more difficult setting as the templates in testing are unseen during training. Intuitively, we would expect a decline in performance as compared to the random split setting. Fig. 2 shows the Macro average F1 scores comparison of *KNN-Former* and multiple baselines under both the random split and the country split. We observe across-the-board performance degradation for all methods after switching from random split to country split. However, the drop is least significant on *KNN-Former*, enabling it to achieve 10% higher F1 than the best baseline. These experiments indicate that our method is more robust and generalizes better to unseen templates as compared to existing baseline models. This is helpful in real-world applications where models frequently encounter new types of documents.

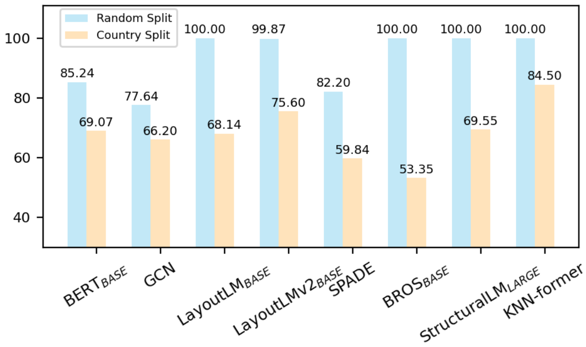


Figure 2: Macro average F1 scores of *KNN-Former* and various baseline models under random split and country split on MIDV2020 dataset.

5.3 Effects of each component in *KNN-Former*

Model	**F1**
KNN-Former	90.76
(-)*KNN* hop attention	88.33 (-2.43)
(-)Local attention based on *KNN* hop & (-)*KNN* hop attention	85.67 (-5.09)
(-)Relative Euclidean distance & angle attention	87.17 (-3.59)
(-)Relative Euclidean distance & angle attention & (-)*KNN* hop attention	86.67 (-4.09)
(-)Combinatorial Matching	88.16 (-2.60)
(+)Absolute positional encoding	86.33 (-4.43)

Table 5: Ablation results on POI dataset. (-) indicates the component is absent compared to *KNN-Former*, (+) indicates the component is additional.

To better understand how *KNN-Former* works, we ablatively study the effects of each component and report the results in Tab. 5. Entity-level detailed results can be found in the Appendix.

Firstly, we observe a 2.43% drop in performance with the removal of *KNN* hop attention and an even bigger 5.09% drop when local attention is removed together with *KNN* hop attention. This demonstrates that the *KNN* graph-based inductive bias is effective in capturing the structural information between document entities. It also shows that local attention, the practice of masking out attention weights when the hop distance between two entities exceeds a pre-defined threshold, further strengthens the inductive bias.

Secondly, we observe that the commonly used spatial inductive bias based on the pairwise relative Euclidean distance and angle also plays an important role. When both relative Euclidean distance attention and *KNN* hop attention are absent, there is a 4.09% drop in performance, an additional decrease of 1.66% compared to when only *KNN* hop attention is ablated(2.43%). The overlap of performance drop suggests some information are captured by both Euclidean distance and hop distance, as some pairs are similarly close/far from each other as measured in both distances. However, each distance also complements the other by capturing additional information. For example, two pairs could carry different importance despite having the same Euclidean distance, due to the presence or absence of other entities positioned between the pairs, signifying the importance of hop distance.

Thirdly, we notice that the F1 score drops drastically by 4.76% when combinatorial matching is ablated. This demonstrates the important contribution of combinatorial matching, as the datasets we experiment on are all subject to a special one-to-one mapping constraint between fields and entities. Combinatorial matching enables our method to treat entity classification as a set prediction problem, instead of predicting each entity's class independently, which enhances our model robustness.

Lastly, we observe that there is a 4.43% drop in performance when absolute positional encoding is added. Previous works (Hwang et al., 2021) have similar findings that adding absolute positional encoding is not helpful, especially when the test set contains a diverse set of unseen templates. In our experiments, adding absolute positional encoding improves performance in training but generalizes poorly in testing.

5.4 Impact of different *K* in the *KNN* graph

To further study the effect of how the hyper-parameter of the *KNN* graph affects the performance, we conduct experiments with different values of *K* on the POI dataset. As shown in Tab. 6, the 2-NN graph achieves the best performance when *KNN*-based hop distance is used and relative Euclidean distance is removed. This is because when only 2 nearest entities are counted as an entity's first-hop neighbors, the correlation between hop distance and entity pair's importance is pronounced. However, a 5-NN graph achieves the best performance when *KNN*-based hop distance is ablated and only relative Euclidean distance is used. This is because the information of who is an entity's 5 nearest neighbors is less useful in documents with an average of 31.79 annotated bounding boxes per file. Models with 2-NN and 5-NN graphs underperform the 4-NN graph in the POI dataset, underscoring the importance of choosing the correct *KNN* graph hyper-parameter for different datasets.

#K	**(+) H (-) R**	**(-) H (+) R**	**(+) H (+) R**
2	90.67	89.33	89.50
5	88.74	90.23	89.51

Table 6: Impact of number of K in *KNN-Former* on POI dataset. (+) indicates presence, (-) indicates absence. H refers to the KNN hop attention. R refers to relative Euclidean distance and angle attention.

5.5 Runtime Comparison

In addition to performance evaluation, we also evaluate the runtime of our model against competitive baselines. For fair comparison, we report the total runtime of sentence transformer plus *KNN-Former*, since *KNN-Former* uses sentence transformer for text embeddings. In fact, the sentence transformer takes up half of the time in our pipeline.

Model	**Single**	**Batch**
LayoutLM$_{\text{BASE}}$	19.61	237.90
LayoutLMv2$_{\text{BASE}}$	56.64	2941.32
SPADE	39.47	6091.52
BROS$_{\text{BASE}}$	23.45	646.65
KNN-Former	22.60	77.57

Table 7: Runtime comparison with baselines. Time taken is reported in milliseconds.

We first measure the runtime to process a single document for each method. As shown in Tab. 7, time taken for sentence encoder plus KNN-former is comparable to LayoutLM and BROS, and is faster than SPADE, LayoutLMv2. We run StruturalLM(written in tensorflow1.14) on CPU due to cuda version mismatch, hence there is no speed measurement.

Moreover, our method allows for significantly larger batch sizes because of the smaller model size. Therefore, runtime for documents in batch is significantly faster than the baselines. Running with maximum possible batch size for each model using a 16GB V100 GPU, *KNN-Former* is significantly faster than the rest, as shown in Tab. 7. This experiment demonstrates that our model is advantageous when faster execution time is desirable, and this could be attributed to the lightweight property of our model.

6 Conclusion

We propose *KNN-Former*, a parameter-efficient transformer-based model for document entity classification. *KNN-Former* uses *KNN* Hop Attention, a new attention mechanism that leverages *KNN* graph-based inductive bias to capture structural information between document entities. *KNN-Former* utilizes combinatorial matching to perform set prediction. We also release POI, a template-rich ID document dataset subject to combinatorial constraints. Experiments show that *KNN-Former* outperforms baselines in entity classification across various datasets.

Limitations

We identify the following limitations in this work. First, the robust performance of baseline methods that leverage image features (Appalaraju et al., 2021) testifies to the importance of visual cues. The inclusion of image features to *KNN-Former* might contribute to better performance. Second, unlike models that perform extensive pre-training (Xu et al., 2020, 2021), *KNN-Former* might lack generic domain knowledge. Third, *KNN-Former* uses a vanilla sentence transformer to get the text embedding inputs. The sentence transformer model is pre-trained and not fine-tuned on the new datasets. An end-to-end training pipeline that jointly trains the text encoding model and *KNN-Former* could lead to better results. Fourth, there are many design choices we did not explore, such as applying attention directly at the token level and pooling representations at the end. Lastly, KNN-Former, along with all baselines used in this work, are subject to OCR failure. All models consume OCR outputs such as bounding box coordinates and texts. In the case of OCR failure, where one bounding box is detected as two or two boxes are merged as one, models that consume OCR results are less likely to make correct predictions.

Ethics Statement

This work has obtained clearance from author's institutional review board. The annotators for POI and MIDV2020 are all paid full-time interns and researchers hired by our institute, whose compensation are determined based on the the salary guidelines of our institute. Among the datasets and annotations released, POI only contains specimens with dummy values, while MIDV is a synthetic dataset. External data are accessed and used in compliance with fair use clauses. We conduct experiments on the private dataset PRV in a secure data zone with strict access control, using auto-labeling scripts for annotations.

Acknowledgement

This research is partly supported by the SRG grant id: T1SRIS19149 and the Ministry of Education, Singapore, under its AcRF Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOET2EP20220-0017). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

References

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.

Vladimir V. Arlazarov, Konstantin B. Bulatov, and Timofey S. Chernov. 2018. MIDV-500: A dataset for identity documents analysis and recognition on mobile devices in video stream. *CoRR*, abs/1807.05786.

Vikraman Arvind, Frank Fuhlbrück, Johannes Köbler, and Oleg Verbitsky. 2020. On weisfeiler-leman invariance: Subgraph counts and related graph properties. *Journal of Computer and System Sciences*, 113:42–59.

Konstantin B. Bulatov, Ekaterina Emelianova, Daniil V. Tropin, Natalya Skoryukina, Yulia S. Chernyshova, Alexander Sheshkus, Sergey A. Usilin, Zuheng Ming, Jean-Christophe Burie, Muhammad Muzzamil Luqman, and Vladimir V. Arlazarov. 2021. MIDV-2020: A comprehensive benchmark dataset for identity document analysis. *CoRR*, abs/2107.00396.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey

Zagoruyko. 2020. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. 2020. Can graph neural networks count substructures? *Advances in neural information processing systems*, 33:10383–10395.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Nima Dehmamy, Albert-László Barabási, and Rose Yu. 2019. Understanding the representation power of graph neural networks in learning graph topology. *Advances in Neural Information Processing Systems*, 32.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2021. BROS: A layout-aware pre-trained language model for understanding documents. *CoRR*, abs/2108.04539.

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.

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 (ICDAR)*, pages 1516–1520. IEEE.

Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. 2021. Spatial dependency parsing for semi-structured document information extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 330–343, Online. Association for Computational Linguistics.

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.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Harold W Kuhn. 1955. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

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. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3735–3754, Dublin, Ireland. Association for Computational Linguistics.

Chenliang Li, Bin Bi, Ming Yan, Wei Wang, Songfang Huang, Fei Huang, and Luo Si. 2021. Structurallm: Structural pre-training for form understanding. *arXiv preprint arXiv:2105.11210*.

Andreas Loukas. 2019. What graph neural networks cannot learn: depth vs width. *arXiv preprint arXiv:1907.03199*.

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

Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. 2019. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609.

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*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. Kleister: Key information extraction datasets involving long documents with complex layouts. In *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part I*, page 564–579. Springer-Verlag.

Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. 2016. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2579–2591, Online. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, KDD '20, page 1192–1200. Association for Computing Machinery.

A Appendix

A.1 Implementation details

We briefly describe the baseline models as well as detailed implemetation details of all models in this section.

- **BERT$_{\textbf{BASE}}$** (Devlin et al., 2019): We use the pre-trained BERT base model for token classification.

- **GCN** (Kipf and Welling, 2016): We use sentence transformer (Reimers and Gurevych, 2019) to get the embeddings of text inputs and use them as the node features for the constructed *KNN* graph. Then we train a 2-layer graph convolutional network to classify the nodes/entities.

- **LayoutLM$_{\textbf{BASE}}$** (Xu et al., 2020): LayoutLM is a transformer-based model for document image understanding. It is pre-trained on IIT-CDIP Test Collection with 11 million scanned images.

- **LayoutLMv2$_{\textbf{BASE}}$** (Xu et al., 2021): In addition to LayoutLM, the LayoutLMv2 adds a new multi-modal task during pre-training to take in the visual cues and incorporates a novel spatial-aware self-attention mechanism.

- **StructuralLM$_{\textbf{LARGE}}$** (Li et al., 2021): On top of LayoutLM, Structural LM uses cell position for each word, and introduces a new pre-training task that predicts the cell position. It is also pre-trained on the IIT-CDIP dataset.

- **SPADE** (Hwang et al., 2021): SPADE builds a directed graph of document entities and extracts and parses the spatial dependency using both linguistic and spatial information.

- **BROS** (Hong et al., 2022): Similar to LayoutLM, BROS is also pre-trained on the IIT-CDIP dataset, but with a different area masking pre-training task, and a different method to encode the 2D positions of bounding boxes.

- **DocFormer** (Appalaraju et al., 2021): DocFormer is a multi-modal transformer that takes in both text and visual cues. It proposes a multi-modal attention mechanism and is pre-trained with several tasks involving both text and image input.

All models are trained on 16G V100 GPUs and implemented with Pytorch, except for StructuralLM$_{\text{LARGE}}$, for which we use their official repository [2] that is implemented in Tensorflow1.14 and we train it on cpu because of cuda version mismatch. We use APIs open-sourced by Huggingface [3] for Bert, LayoutLM$_{\text{BASE}}$ and LayoutLMv2$_{\text{BASE}}$. SPADE is implemented using the official implementation released by ClovaAI[4]. BROS is implemented using their released official repository [5]. Only text inputs are passed to BERT$_{\text{BASE}}$ for classification while bounding box coordinates are neglected. Results are obtained after training for 100 epochs. We trained the SPADE

[2] https://github.com/alibaba/AliceMind/StructuralLM

[3] https://huggingface.co

[4] https://github.com/clovaai/spade

[5] https://github.com/clovaai/bros

model for 10 to 20 hours up to 1000 epochs depending on the datasets. All settings of LayoutLM$_{\text{BASE}}$ and LayoutLMv2$_{\text{BASE}}$ are from the authors. For BROS, we use the same tokenizer as LayoutLM, same learning rate in their paper and fine-tuned BROS on each dataset for at least 100 epochs, and made sure it converged. We report results for epoch 80. For StructuralLM$_{\text{LARGE}}$, we were only partially successful to reproduce it due to OOV error when running on POI dataset. In addition, this is the only baseline that we use the large version because there was an error with the base version. we train the model with 25 epochs with all other hyperparameters following their paper. We reproduced DocFormer from an unofficial repository [6] since there is no official repository available. There is no released pretraining weights for DocFormer, but DocFormer uses plain ResNet50 (He et al., 2016) as the first step for image feature extraction, and the language embedding weights are initialized with LayoutLMv1$_{\text{BASE}}$ pre-trained weights. We trained DocFormer for at least 100 epochs and used hyperparameters for fine-tuning setting mentioned in the paper. We report results for epoch 100.

For *KNN-Former*, we use 8 layers, 8 heads, and 80 hidden dimensions for the architecture. Results are obtained after training for 400 epochs. We use a 6-layer sentence transformer to extract text features in for both *KNN-Former* and GCN baseline model implementation. We use Adam optimizer with learning rate of 5e-3. We perform a grid search in choosing hyper-parameters, with learning rate in [5e-3, 1e-3, 5e-4], the number of layers in [4, 8], local attention threshold in [1,2,3], and the number of attention heads in [4,8]. To incorporate relative Euclidean distance and angle, we tried both real and quantized angles in our initial exploration and did not find a significant difference. We use real angle values throughout the experiments. In the implementation of combinatorial matching, we choose class probabilities as matching cost following (Carion et al., 2020). Despite no theoretical justification, they observe better performance than log probabilities. We conduct experiments comparing class and log probabilities but do not observe significant differences in POI dataset(<0.005%). Reported results are the average performance of 3 runs. The sentence transformer we used is paraphrase-MiniLM-L6-v2 from hugging face.

A.2 Experimental Results on MIDV2020 random split

Tab 8 shows the additional experimental results on MIDV2020 random split. Column L.Name, F.Name, DoB, DoI, DoE and ID No. correspond to results of Last Name, First Name, Date of Birth, Date of Issue, Date of Expiry, and ID Numbers. GCN and *KNN-Former* have additional 22.7 M fixed parameters since we employed a lightweighted 6-layer sentence transformer (Reimers and Gurevych, 2019) to get the text embeddings. MIDV dataset has 10 templates, and each template has 100 images. As a result, this random split is an easy setting where performance results are generally good. BERT$_{\text{BASE}}$ still produces relatively poor performance, which reiterate the point that spatial information is important.

A.3 Experimental Results on DocFormer

Tab 9 shows the experimental results of DocFormer on various datasets. On POI, PRV dataset and MIDV2020 dataset random split, DocFormer performs reasonably well. On POI dataset, it only falls behind LayoutLMv2$_{\text{BASE}}$ and *KNN-Former*; on PRV dataset, it outperforms BERT$_{\text{BASE}}$, GCN and SPADE; on MIDV2020 dataset random split, it achieves 100% F1 score for every field like *KNN-Former*, StructuralLM$_{\text{LARGE}}$, LayoutLM$_{\text{BASE}}$ and BROS$_{\text{BASE}}$. However, on MIDV2020 dataset country split, we cannot get reasonable performance for DocFormer although we made sure our training was converged.

We also measured the runtime of DocFormer, results shown in Tab. 10.

A.4 POI Dataset Details

All images are publicly available specimen ID documents and do not contain information about real persons. Despite that, due to the sensitivity of the subject and increasing societal concerns about the role artificial intelligence should play in protecting people's privacy, we will only release the annotated JSON file instead of the actual images to comply with fair use of specimens.

We store a list of objects in the annotated file; each object contains annotations for an image. The annotations include bounding box coordinates, text, and category.

The released dataset is subject to fair use clause and should only be used for academic purposes.

We implement quality control during the annota-

[6] https://github.com/shabie/docformer

Dataset	Method	F1 Score						Trainable Param
		L.Name	F.Name	DoB	DoI	DoE	ID No.	
MIDV	BERT$_{\text{BASE}}$	72.09	81.35	100.00	92.99	88.48	76.52	110 M
	GCN	51.48	61.66	98.68	91.59	88.55	73.90	31.5 K
	StructuralLM$_{\text{LARGE}}$	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	355M
	LayoutLM$_{\text{BASE}}$	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	110 M
	LayoutLMv2$_{\text{BASE}}$	99.47	99.74	**100.00**	**100.00**	**100.00**	**100.00**	199 M
	SPADE	88.14	86.82	70.63	80.33	79.71	87.55	128 M
	BROS$_{\text{BASE}}$	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	109 M
	KNN-Former	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	0.5 M

Table 8: Experimental Results on MIDV2020 Random Split.

Dataset	Method	F1 Score						Input Modality	#Parameters	
		L.Name	F.Name	DoB	DoI	DoE	ID No.		Trainable	Total
POI	DocFormer$_{\text{BASE}}$	78.22	78.87	95.15	90.99	91.82	81.65	text + layout + image	110M	110M
PRV		78.21	84.86	98.17	96.42	97.38	91.89			
MIDV2020 (random split)		100.00	100.00	100.00	100.00	100.00	100.00			
MIDV2020 (country split)		1.50	0.00	0.00	1.91	0.00	0.00			

Table 9: Experimental Results on DocFormer.

Model	Single	Batch
LayoutLM$_{\text{BASE}}$	19.61	237.90
LayoutLMv2$_{\text{BASE}}$	56.64	2941.32
SPADE	39.47	6091.52
BROS$_{\text{BASE}}$	23.45	646.65
DocFormer$_{\text{BASE}}$	71.57	7485.10
KNN-Former	22.60	77.57

Table 10: Runtime comparison with baselines. Time taken is reported in milliseconds.

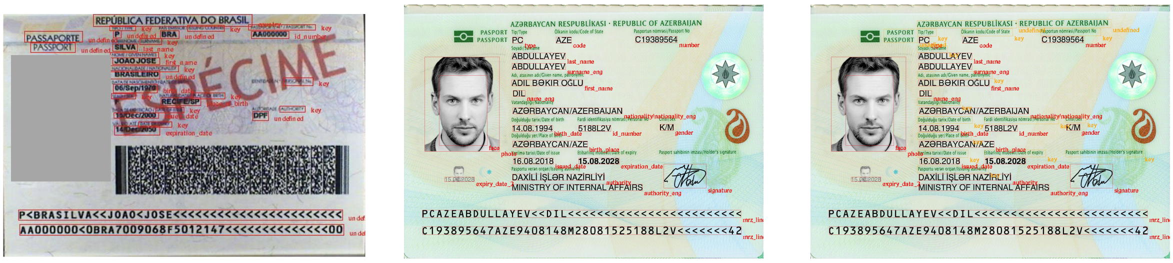
tion process by having annotators cross-check each other's results to affirm the correctness of labels.

A.5 Sample documents of POI and MIDV2020

In Fig. 3, we show samples documents with bounding boxes and annotations.

A.6 PRV Dataset Details

Since POI and MIDV2020 only contain specimens or artificially generated images, we run our model on a private (PRV) dataset that consists of actual ID documents. The documents are protected by strict privacy requirements and massive human annotations are not available as raw images are inaccessible. Therefore, we build automatic labeling to annotate the ground truth. Specifically, we map personal information in the existing database to OCR-ed text outputs. The matched bounding box is classified as the corresponding entity if a match is found. All bounding boxes that are not matched are classified as 'others'.


REPÚBLICA FEDERATIVA DO BRASIL
PASSAPORTE
PASSPORT
ESPÉCIME
AZƏRBAYCAN RESPUBLİKASI · REPUBLIC OF AZERBAIJAN
ABDULLAYEV
ADIL BƏKIR OĞLU
AZƏRBAYCAN/AZERBAIJAN
DAXİLİ İŞLƏR NAZİRLİYİ
MINISTRY OF INTERNAL AFFAIRS

(a) POI document

(b) Original MIDV2020 document

(c) Enhanced MIDV2020 document

Figure 3: Example documents with bounding boxes and annotations. There is only one entity box corresponding to one field of interest.