One-Shot Doc Snippet Detection: Powering Search in Document Beyond Text

Abhinav Java*, Shripad Deshmukh*, Milan Aggarwal, Surgan Jandial, Mausoom Sarkar, and Balaji Krishnamurthy

Adobe Media and Data Science Research Labs, Noida, India
{ajava, shdeshmu, milaggar, jandial, msarkar, kbalaji}@adobe.com

Abstract

Active consumption of digital documents has yielded scope for research in various applications, including search. Traditionally, searching within a document has been cast as a text matching problem ignoring the rich layout and visual cues commonly present in structured documents, forms, etc. To that end, we ask a mostly unexplored question: "Can we search for other similar *snippets present in a target document page given a single query instance of a document snippet?". We propose* MONOMER *to solve this as a one-shot snippet detection task.* MONOMER *fuses context from visual, textual, and spatial modalities of snippets and documents to find query snippet in target documents. We conduct extensive ablations and experiments showing* MONOMER *outperforms several baselines from one-shot object detection (BHRL), template matching, and document understanding (LayoutLMv3). Due to the scarcity of relevant data for the task at hand, we train* MONOMER *on programmatically generated data having many visually similar query snippets and target document pairs from two datasets - Flamingo Forms and PubLayNet. We also do a human study to validate the generated data.*

1. Introduction

Documents have been the primary medium for storing and communicating information in academia, public offices, private businesses, print media, etc [23]. With world transitioning to a digital-first ecosystem, expedited by the challenges posed by the ongoing pandemic [1], the trends in document usages are shifting from passive modes like reading/sharing to more active modes such as authoring a document, editing the styles, customising tables, etc. However, search functionality within documents is mostly limited to locating regions in a page containing text that matches a given textual query [6, 9, 35]. Confining search to textual modality restricts several use cases. For instance, consider scenarios where a user wants to search for border-less tables to add borders while editing the document, or an author of loan form wants to search for pattern containing binary gender options to look surrounding contextual fields in personal detail sections, or a document editor wants to search for image-caption pairs to swap their ordering. These scenarios emphasise the need for more advanced search capabilities based on document snippets.

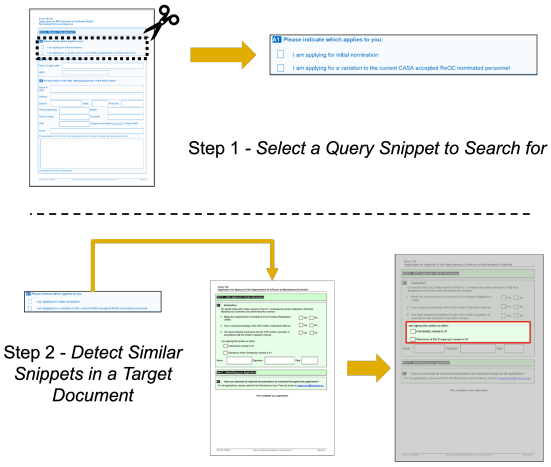

Figure 1: A new paradigm for search in documents through one-shot snippet detection.

Hence, a utility that allows selecting a rectangular snippet in a page, and find other similar snippets in a target document would be a stepping stone towards empowering this search experience. To achieve this, we model it as *one-shot doc snippet detection* task i.e. detect regions in a target document page that are similar to a given snippet (as shown in Fig. 1). Existing text-based search tools [13] are incapable of detecting visually similar snippets as they lack mechanism to incorporate visual and layout cues. On the other

*These authors contributed equally to this work

hand, document structure extraction methods [3, 21, 38] are trained to identify predefined generic class structures (such as a paragraph) in a document and hence, cannot be applied directly to detect arbitrary snippet patterns. Further, document image retrieval tasks such as logo-detection [4], signature field retrieval [17] etc. are designed to extract task-specific entities like logos and sign-fields respectively.

To attain the capability of "search with snippets" in documents, we formulate the problem as a one-shot snippet detection task and design a multi-modal method for the same. We propose a multi-modal framework – ***MONOMER*** (**M**ultimodal Cr**O**ss Atte**N**tion-based D**O**cu**M**ent Snipp**E**t Sea**R**ch) that fuses context from visual, textual and spatial modalities across the query snippet and target document through attention (Section 4). The fused representations are then processed through a feature pyramid network, followed by region proposal predictions, to identify the boundaries of regions in the target document that are similar to the query snippet. We compare our approach with the current state-of-the-art method in One-Shot Object Detection - BHRL [39] and a task-specific extension of the best performing document analysis approach - LayoutLMv3 [16] in Table 2. We show that MONOMER outperforms the above baselines (Section 5.4), highlighting the effectiveness of our proposed framework. In Section 5.5 we demonstrate the advantage of using all three modalities by performing extensive ablations with various modality combinations.

The scarcity of relevant data poses an additional challenge in tackling the task. Documents in the form of images, text, and layout[8] are widely available. However, annotated data of document snippets and their associated matching regions in other documents are hard to find. Additionally, different modalities like visual, layout and textual imply similarity in a highly subjective manner. This makes obtaining large-scale human-annotated data for snippet search extremely challenging. To make the problem tractable, we design a programmatic way of obtaining similar query snippet and target document pairs by defining similarity based on alignment between the layout of their basic constituent structures. More specifically, we sort the constituent structures (such as Text, Tables, Fillable areas etc.) in a doc snippet according to natural reading order, followed by creating a layout string based on the sorted sequential ordering. Likewise, we obtain the layout string corresponding to each document in the corpus. The snippet is associated with documents whose layout string has at least one contiguous subsequence that aligns with the snippet's layout string. Therefore, we propose a layout-centric definition of similarity that enforces alignment between the layout of two snippets to be deemed similar. We choose Flamingo forms [34] and PubLayNet documents [43] as the underlying corpora to create two similarity matching datasets. We discuss the data generation procedure in detail, followed by its validation through human study in Section 3.2. To summarize, our contributions can be enumerated as the following:

- We formulate the task of one-shot document snippet detection for advancing search in document domain beyond traditional text-based search.
- We define layout based document-snippet similarity that allows generation of similarity matching data at scale in a fully programmatic manner, the validity of which is supported by an extensive human study. We plan to release a part of the introduced datasets.
- We propose MONOMER, a multi-modal framework for snippet detection that performs better than one-shot object detection and multi-modal document analysis baselines. Further, MONOMER is able to perform well on layout patterns not seen during training.

2. Related Work

2.1. Document Understanding

Understanding documents requires comprehending the content present in document page i.e. images, text, and any other multi-modal data in conjunction with the layout, structures, placement of the content, blank spaces, etc. For understanding content, prior research works have designed tasks such as DocVQA [35], InfographicsVQA [24] etc. while layout understanding has been formally studied through Document Layout Analysis [2, 7]. Layout analysis has been formulated as an object detection task [42] to extract structures such as headings, tables, text blocks, etc. from a document image. Such approaches extensively use state-of-the-art object detection heads (for eg. YOLO [29], Faster-RCNN [31] etc.) usually employed in the domain of natural images. Methods such as HighResNet [34], MFCN [40] etc. approach Layout Analysis as pixel-level segmentation of document image. Subsequently, several recent works like DocFormer [5], LayoutLM [38], DiT [19] proposed large-scale pre-training techniques to cater the document understanding task. The representations learned by these models have turned out to be very useful in many downstream tasks, both for content understanding and layout parsing. In this work, we leverage such representations to develop snippet based search tools for the documents.

2.2. Template Matching

Template Matching refers to the task of detecting and localizing a given query image in a target (usually larger) image. Seminal template matching literature leverages traditional computer vision techniques like Normalized Cross Correlation (NCC) [41] and Sum of Squares Differences (SSD) [14] for searching. Despite their widespread success, the aforementioned techniques have clear limitations in regard to matching templates which are complex transformations of the instance present in the target image. For in-

stance, NCC/SSD might fail due to large variation in scale, occlusions etc. Consequently, feature matching-based techniques such as SIFT [37] and SURF [28] were proposed to allow matching local features between images to address scale-invariance. Typically, these methods find local keypoints in images. Yet, several issues like image quality, lightning, real-time use severely limit the applicability of these approaches. The recent surge of Deep Learning allowed researchers to develop more sophisticated techniques like QATM [11], DeepOneClass [33] that perform Siamese matching between deep features of natural images for tasks like GPS localization. QATM [11] propose a learnable matching layer that enables better matching within natural images compared to standard Siamese matching. However, we note that matching templates within documents is a different (from natural images) and non-trivial task with additional nuances owing to the diverse and complicated arrangement of layout, visual structures and textual content contained in a document.

2.3. One Shot Object Detection (OSOD)

OSOD aims at detecting instances of novel classes (not seen during training) within a test image given a single example of the unseen/novel class. At a high level, most OSOD techniques perform alignment between deep features of query (example of novel class) and target image (test image where the novel class instance is present). Recently, methods such as COAE [15], AIT [10] etc. have shown that the learned attention-based correlation can outperform standard Siamese matching [18, 25] since they capture multi-scale context better through global and local attention. Popular OSOD techniques [22] have been shown to perform well on natural images when class definitions are clearly specified. However, due to the complexity of document data and lack of a well-defined yet exhaustive set of layout patterns, it is not possible to enumerate a finite set of classes. More recently, [39] proposed a technique to learn a hierarchical relationship (BHRL) between object proposals within a target and the query. While BHRL shows impressive performance on natural images, it does not leverage multi-modal information that is critical for document snippet detection. Contrary to existing approaches, we leverage all possible co-relations between different modalities of query and target and show that we are able to achieve better overall performance on complex document data where existing methods typically fail.

3. One-Shot Doc Snippet Detection

3.1. Problem Formulation

We first give an overview of dataset creation and outline of the task formulation followed by their details.

Dataset Creation. Let $\mathcal{X}$ be the set of all document snippets. We define a similarity criterion $g_{sim} : \mathcal{X}^2 \rightarrow \mathcal{R}$ which takes two document snippets $A, B \in \mathcal{X}$, and outputs similarity score $s = g_{sim}(A, B)$. The g_{sim} function can be defined according to human's notion of similarity, or as a fully programmatic similarity criterion. Query-target pairs (Q, T) are mined from the document corpus using g_{sim} such that $Q \in \mathcal{X}$ and target document T contains a non-empty set of snippets $\mathcal{S}_{qt} = \{S_i | S_i \in \mathcal{X}, g_{sim}(S_i, Q) > th_{sim}, i = 1, 2, ..., n\}$; th_{sim} being the threshold over similarity score. (Q, T) pairs are collected to create dataset $\mathcal{D}$.

Task Definition. Given a dataset $\mathcal{D}$ of query-target pairs which are generated using an oracle g_{sim} (not accessible afterwards), our task is to find $\mathcal{S}_{qt}$ for each pair $(Q, T) \in \mathcal{D}$. Let f_θ be a model with parameters θ which predicts similar snippets $\hat{S}_{qt}$ for given (Q, T) pair and let loss L be the measure of error between S_{qt} and $\hat{S}_{qt}$. Then the doc snippet detection task becomes that of minimizing L as follows

$$\min_{\theta} \sum_{\forall (Q,T) \in \mathcal{D}} L(\mathcal{S}_{qt}, \hat{\mathcal{S}}_{qt})$$

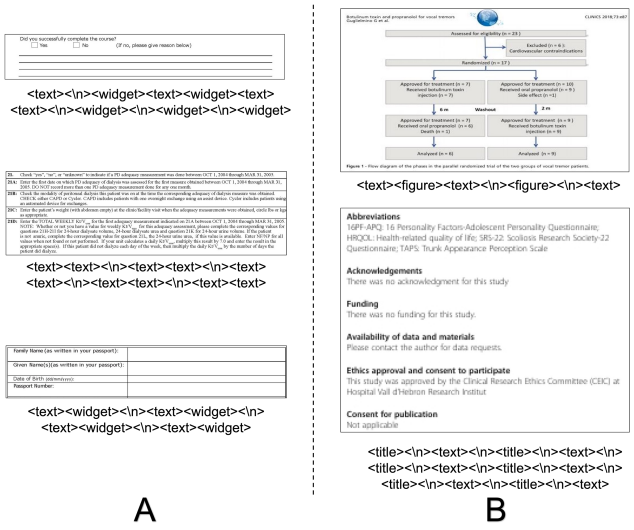

Figure 2: Snippets and the corresponding layout strings for A) Flamingo Forms, B) PubLayNet Documents.

3.2. Snippet-Document Dataset

In this section, we discuss the details of how we define similarity in the context of documents using the layout of different snippets and documents to generate $\mathcal{D}$ followed by a human study to validate the quality of generated data.

3.2.1 Dataset Generation

Since document similarity depends on various factors and is highly prone to subjectivity, obtaining significant number of (Q, T) pairs through human annotation becomes quite

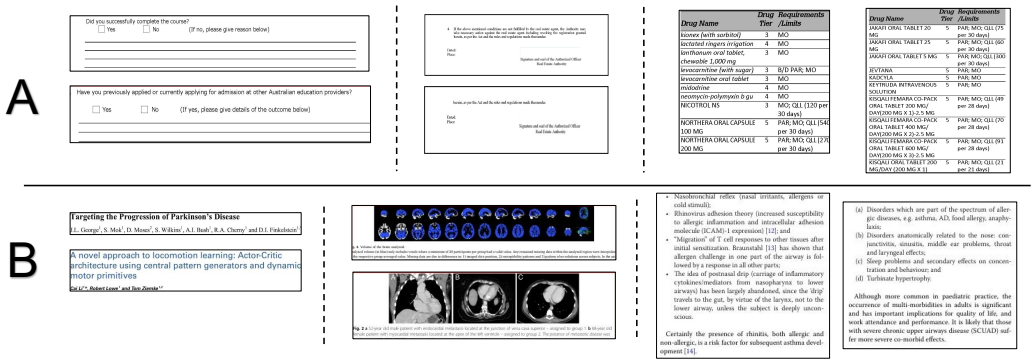

Figure 3: Similar Snippets extracted programmatically for - A) Flamingo Forms, and B) PubLayNet Documents.

challenging. To that end, we decide to define g_{sim} criterion programmatically as the following:

$$g_{sim}(A, B) = 1 - \frac{d(lstr_a, lstr_b)}{length(lstr_a)} \quad (1)$$

where $lstr_a$ and $lstr_b$ denote the layout strings of snippets A and B respectively and d denotes the edit distance [26]. To obtain the layout string of a snippet or full document page, we sort their constituent structures (such as Text, Tables, Fillable areas etc.)[1] according to natural reading order (top-bottom and left-right). We associate a symbol with each constituent element type such that the sequence of element symbols obtained according to the sorted ordering yields the layout string. Fig. 2 shows examples of snippets and their corresponding layout strings. Given a snippet $\mathcal{Q}$ extracted randomly from some document, we provide its layout string as argument $lstr_a$ in Eq. 1. To identify if some other document in the corpus contains a similar region, we consider all possible contiguous subsequences of its layout string as candidates and provide the subsequence as input $lstr_b$ in Eq. 1. We filter candidates which qualifies the similarity threshold th_{sim} of 0.92 (determined based on observation) to generate query-target pairs.

Size and spacing: Additionally, to address the issues related to span and vertical & horizontal spacing, we apply a size filter to the positives extracted by Eq. 1, that ensures which the size of positive regions in the target is similar (within a threshold) to that of the query snippet. Further, it is desirable not to overfit on size and allow some permissible variation between query and target region since we want them to be structurally and visually similar but not exactly the same. This allows incorporating minor variations for the same layout with respect to scale and relative arrangement of constituent elements. Lastly, in the case of forms, blank spaces are mostly present in the form of fillable areas i.e. "widgets" which we are taken into account while creating the layout string for matching. Fig. 3 illustrates the similar snippets identified using proposed g_{sim}.

Dataset	*No. of (Q, T) pairs*		*Unique Layout Strings*	
	Train	Test	Train	Test
Flamingo	102065	24576	6365	1911
PubLayNet	204256	15734	35	23

Table 1: Summary of snippet-document pairs datasets. Less number of unique layout strings in PubLayNet indicates limited combinations in which structures are organised.

Base datasets: We derive two similarity search datasets from two multi-modal document corpora respectively – the Flamingo forms dataset [34] and the PubLayNet document dataset[2] [43]. The rationale behind choosing them is, a) forms data contain diverse layout structures with various hierarchical levels [3, 34], and b) PubLayNet is a commonly used large scale documents dataset for document analysis. For Flamingo dataset, we use widgets (fillable area) and text blocks to create layout strings and for PubLayNet we consider text blocks, figures, lists, tables and titles as the layout symbols. Table 1 summarises number of samples obtained. We release our dataset here.

Rationale for visual and layout similarity: We do not consider the text when estimating edit distance to avoid situations where a target document contains a region with text similar to the query snippet but with a very different structure. Specifically, consider the case in which the text is similar in a paragraph and a table—modifying our dataset generation method on text similarity could result in labeling two dissimilar structures as similar. To avoid that, we limit the scope of this work to visually similar regions where text may or may not be similar.

Limitations of data creation heuristic: For details and ex-

[1]The bounds for basic elements are either present in the dataset or can be extracted using auto-tagging capabilities of PDF tools.

[2]https://developer.ibm.com/exchanges/data/all/publaynet

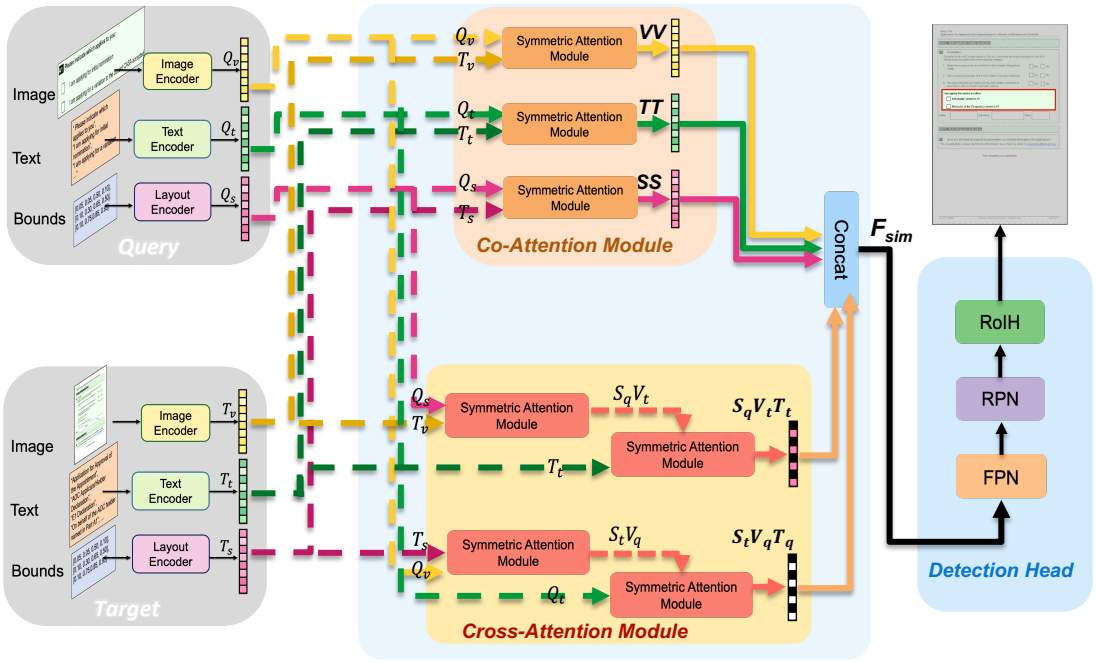

Figure 4: Architecture of the proposed MONOMER approach.

amples, please refer to supplementary (Section 4).

3.2.2 Human Study on Generated Data

To validate the quality of the generated data, i.e. to evaluate how well the programmatically generated query-target pairs align with human notions of similarity, we conduct a human study involving 12 evaluators[3]. We evaluate a total of 160 snippet-target document pairs sampled randomly from our dataset generated using Forms such that these samples are divided into 4 batches of 40 samples each. All samples in a batch are then evaluated by 3 evaluators based on the following criteria - given regions in a target document, count the number of regions 1) that are highlighted as similar and are actually similar, 2) that are similar but not highlighted, 3) which are highlighted as similar but are not exactly the same as the snippet. The evaluators are also asked if the layout pattern of the snippet is hard. Based on above, we estimate batch-wise metrics such as precision, recall etc. and report the average across the batches[4]. It is found that precision is 87.96% i.e. in $\sim$ 88% cases, target document snippet highlighted as similar to query snippet by our method is actually similar; recall is found to be 81.07% which indicates that $\sim$ 81% of actually similar regions are highlighted by our method. Further, it is found that 87.48% of similar matches are the ones where target document region is not exactly the same as the snippet showing that our technique mostly identifies similar but not trivially exact matches. Finally, it is observed that 48.12% of snippets comprise of layout pattern which is complicated and hard to search.

Link with performance on real data: For discussions and experiments on comparison of our method with data creation heuristic as baseline on human annotated data, refer to supplementary (Section 3).

[3]Evaluators were remunerated appropriately for the evaluation task

[4]Please refer to supplementary (Sec. 2) for batch-wise details

4. MONOMER

As information in a document is mainly present in the form of images, text and layout, paradigms that leverage all the modalities simultaneously have turned out to be successful. For instance, document analysis methods such as DocFormer [5], SelfDoc [20], LayoutLMv3 [16] etc. have developed pre-trained multi-modal architectures achieving great results on a wide variety of tasks like layout extraction, text recognition, document image classification, etc. Motivated by this, we design our framework with the aim of enabling it to pool context from various document modalities to perform one-shot snippet detection task.

A possible way to leverage multi-modal context is to directly use one of the aforementioned pre-trained models to obtain multi-modal embeddings for both query snippet and target document separately. However, doing so restricts interconnecting individual modalities between query snippet and target page. We substantiate this intuition empirically in Table 2 by comparing our method against fine-tuning pre-trained multi-modal baselines for doc snippet detection.

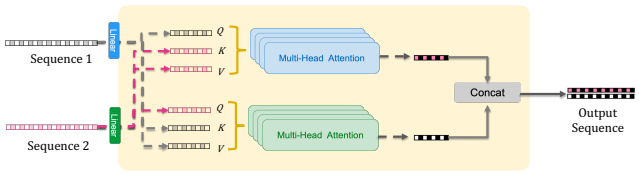

Figure 5: Architecture of symmetric attention module (zoom in for better view).

Hence, we embed each modality for both snippet and target document separately using image, text and layout encoders and process them further through interconnecting attention between modalities. We now discuss our architecture (Fig.4) in detail.

Feature Extraction. To encode the snippet and target document image, we use DiT [19], a visual-only document analysis model. The text present in query and target is encoded using BERT [12] based text encoder. Additionally, we generate features for bounding box information of constituent elements of snippet (arranged in a sequence according to reading order). Specifically, we use a transformer [36] based encoder-only module which tokenizes box coordinates and embeds the sequence of bounds. Previously, methods like [30, 38] have used such an approach to process various types of sequential data. Consequently, we generate 6 types of embeddings in total (3 modalities of the query and 3 modalities of the target). The visual, textual and spatial embeddings for query snippet are denoted by Q_v, Q_t and Q_s respectively and likewise, for the target document we have T_v, T_t and T_s.

Parenthetically, the visual and spatial embedding of a document are highly interconnected as bounding boxes of documental entities determine the visual outlook of a page. Also, the visual embeddings from a vision-only model like DiT have demonstrated ability to detect and recognize text in downstream tasks [19], implying that the visual features also contain information about the text content of the document. Building on this intuition, we combine the extracted features accordingly.

Query-Target Feature Fusion Features Q_v, Q_t, Q_s, T_v, T_t, and T_s are in the form of token sequences outputted by corresponding transformer based encoders. We strategically apply *symmetric attention* [10] between these token sequences. A symmetric attention of two sequences involves i) computing multi-head attention [36] of first sequence as query with the second sequence as key and value, ii) computing multi-head attention of the second sequence as query, and first sequence as key and value, iii) concatenating the attention outputs along feature axis to obtain final sequence output. The same is depicted in Fig. 5.

We apply *co-attention* (i.e. symmetric attention between sequences of *same* modalities) between Q_v & T_v, Q_t & T_t and Q_s & T_s to generate output sequences VV, TT and SS respectively. VV, TT and SS contain information about the correlation between the query and target features of the same modality.

Building on our starting intuition regarding interconnection between different modalities, we first compute *cross-attention* (i.e. symmetric attention between sequences of *different* modalities) between $Q_s - T_v$ and $T_s - Q_v$ to generate *spatio-visual* embeddings S_qV_t and S_tV_q. As mentioned earlier, T_v contains information about T_t and likewise, same is the case with Q_v and Q_t. Therefore, to leverage these relations, we compute cross-attention of S_qV_t and S_tV_q with T_t and Q_t respectively. Finally, we get *spatio-visio-textual* encodings $S_qV_tT_t$ and $S_tV_qT_q$.

Detection of Similar Snippets. Finally, we have 5 token sequences (each with max length and feature dimension as 1024) – 3 Co-Attention sequences: VV, SS, TT and 2 Cross-Attention sequences: $S_qV_tT_t$ and $S_tV_qT_q$. These sequences are simply concatenated along the last dimension to form a feature volume $F_{sim} \in \mathcal{R}^{BS\times1024\times5120}$, where BS represents the size of the batch and 1024 is the maximum sequence length (hyperparameter). We posit that this feature volume contains all the necessary information to find the relevant snippets within the target. We apply a linear projection on F_{sim} and convert it to a vector of shape $BS \times 1024 \times 4096$, that is reshaped into a feature volume $F_{feat} \in \mathcal{R}^{BS\times1024\times64\times64}$.

A sequence of conv layers, each with a kernel size of 1, followed by LeakyReLU activation (slope$= 0.1$), processes F_{feat} to output features at 4 different levels, with shape - $BS\times256\times64\times64$, $BS\times512\times64\times64$, $BS\times1024\times64\times64$, $BS \times 2048 \times 64 \times 64$. The hierarchical features are subsequently processed through a standard FPN architecture, followed by the FasterRCNN RPN and RoI heads [31] to obtain the final bounding boxes. Please refer to the supplementary for further details about the FPN and RPN modules, hidden dimension of other modules, size of intermediate vectors obtained through attention, etc.

5. Experiments and Analysis

5.1. Implementation Details

We train MONOMER using standard Object Detection losses i.e proposal matching + bounding box (used in Faster-RCNN) [31] on a batch size of 48 (6 per GPU, total of 8 GPUs). Optimization is performed using SGD [32] with momentum 0.9 and weight decay $1e-2$. The initial learning rate is set to $5e-2$ and updated with a cosine annealing scheduler. The output of detection head is processed with a confidence threshold of 0.4 on prediction and NMS [27] threshold of 0.45 on IoU. For all the experiments, we uniformly use 8 Nvidia A-100 GPUs.

Model	*Flamingo Forms*					PubLayNet Documents				
	AP50	AP75	AR50	AR75	mAP	AP50	AP75	AR50	AR75	mAP
SSD	-	-	0.0000	0.00	0.00	-	-	0.01	0.00	0.00
NCC	29.41	24.82	5.16	0.00	2.77	46.09	29.94	18.60	0.04	7.36
BHRL (CVPR'22)	58.09	51.00	38.67	30.28	35.45	36.74	26.18	54.55	28.69	22.47
LayoutLMv3 (MM'22)	51.45	43.21	**58.88**	38.80	45.51	35.95	16.50	65.38	18.31	21.46
MONOMER (Ours)	**78.16**	**73.93**	56.65	**51.11**	**66.95**	**64.30**	**39.83**	**64.18**	**32.95**	**36.61**

Table 2: Comparing MONOMER's performance against other approaches at the task of one-shot doc snippet detection. (Note: hyphen denotes that no boxes were detected, same is reflected in the mAP and recall.)

MONOMER Variant	*Flamingo Forms*					PubLayNet Documents				
	AP50	AP75	AR50	AR75	mAP	AP50	AP75	AR50	AR75	mAP
Image	67.33	62.40	**59.49**	49.95	63.73	53.43	30.13	60.29	23.98	23.75
Image + Text	72.46	67.60	57.97	50.25	64.31	62.08	36.53	58.03	27.27	33.00
Image + Bounds	70.37	65.50	57.30	49.06	63.30	57.67	34.10	**69.21**	32.33	32.91
Image + Text + Bounds	**78.16**	**73.93**	56.65	**51.11**	**66.95**	**64.30**	**39.83**	64.18	**32.95**	**36.61**

Table 3: Analysing performance of MONOMER variants that use different combinations of modalities.

5.2. Baselines

We begin with applying standard template matching approaches: Normalized Cross-Correlation (NCC) and Sum of Squared Differences (SSD) to detect similar snippets. Further, owing to resemblance of the proposed task's with one-shot object detection (OSOD) setting, we train BHRL[5], the current state-of-the-art in OSOD. Additionally, we implement an approach using top-performing document analysis model LayoutLMv3[6], where the query-target are embedded separately to generate multi-modal features that are processed through symmetric attention and detection head. We add details of model sizes compared with aforementioned baselines in the supplementary material (Sec 1.4).

5.3. Evaluation Metrics

We adopt the metrics from one-shot object detection for evaluating performances of various approaches on the one-shot doc snippet detection. Specifically, we measure Average Precision(AP) and Average Recall(AR) at IoU thresholds of 0.50 and 0.75 which are denoted by AP_{50}, AP_{75}, AR_{50} and AR_{75} respectively. In addition, we calculate mean Average Precision (mAP) [22] of the predictions by averaging APs at IoU thresholds starting from 0.50 and increasing in the steps of 0.05 till 0.95.

5.4. Results

Table 2 shows the results of different approaches at doc snippet detection task. We see that the template matching algorithms perform very poorly on this task, the reason being their inability to adapt to the transformations in the similar snippets such as aspect ratios, font sizes, styles, and the like. BHRL shows significant improvement over template matching, but its performance plateaus early because of lack of understanding of the text and layout information in the documents. LayoutLMv3, with its rich document representations, demonstrates improvement over aforementioned techniques. Using multi-modal embeddings directly as in the LayoutLMv3's extension, doesn't provide an explicit control over individual modalities of the query and the target. MONOMER, with more flexibility to process information streams, gives better mAP in both data settings.

Qualitative Visualizations. We discuss the key differences between the qualitative outputs produced by MONOMER against other strong baselines. A summary of the results is illustrated in Fig. 6. The query snippets contain certain layout patterns whose corresponding matches in the target document are shown by the green bounds in ground truth columns. As we can observe, the query is not *exactly* the same as regions marked in the ground truth, thus making the detection task non-trivial. We note that MONOMER is able to detect several complex patterns in the form, clearly performing better than the detections made by the baselines. For instance, the top-left (row 1, Flamingo-Forms) example in Fig. 6 demonstrates that while BHRL is able to detect most of the true positives, it also detects two regions as false positives. We attribute this behavior of BHRL to its over-reliance on a limited number of classes (all choice-like patterns are detected instead of the similar layout). Further, LayoutLMv3 also predicts a number of extraneous bounding boxes that do not match the ground truth. Similarly, at the bottom-left

[5] https://github.com/hero-y/BHRL

[6] https://huggingface.co/docs/transformers/model_doc/layoutlmv3

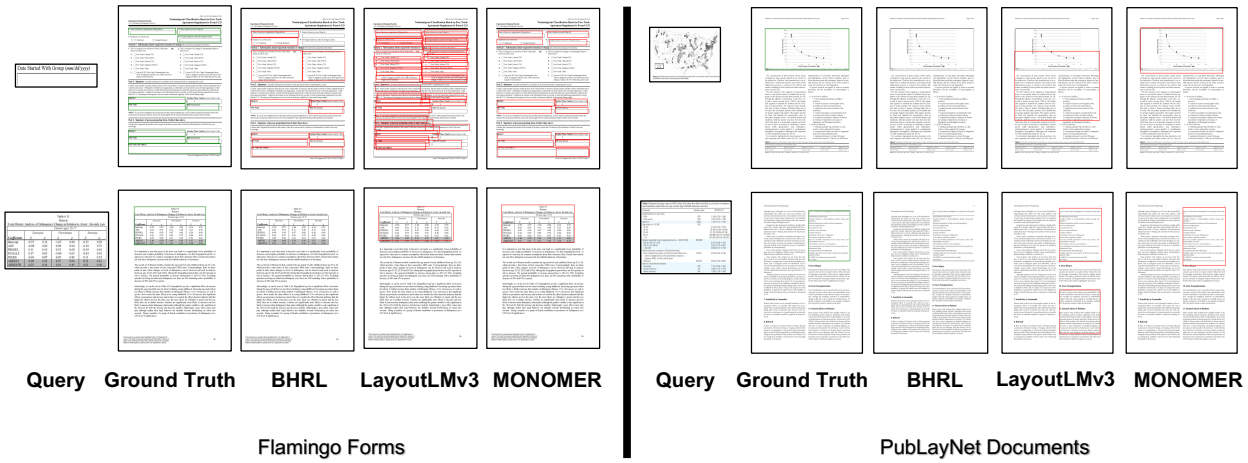

Figure 6: Qualitative comparison between BHRL, LayoutLMv3 and MONOMER. (Zoom in for better view)

in Fig. 6 (row 2, Flamingo Forms) the superior precision of MONOMER over both LayoutLMv3 and BHRL can be observed. Furthermore, MONOMER yields better quality detections even in the PubLayNet dataset as shown in Fig. 6 (right). We note both BHRL and LayoutLMv3 often fail to predict bounding boxes for examples in the PubLayNet dataset, whereas MONOMER consistently predicts them. The efficacy of our method over LayoutLMv3 and BHRL can be observed in Fig. 6 (row 2, PubLayNet) wherein LayoutLMv3 produces a false positive and BHRL does not yield any prediction. Please refer to supplementary for more qualitative analysis.

5.5. Ablation And Analysis

Performance on varying Modalities. We quantify the roles played by individual document modalities in MONOMER's performance through an ablation study where we switch the modality information on/off in different combinations. First, we consider image-only variant of MONOMER. To this model, we add text and bounding box modalities separately to get two more MONOMER variants. Table 3 compares performances of these variants against MONOMER trained on image, text and bounds together. Model processing all modalities surpasses other variants significantly, underlining the usefulness of incorporating document-specific nuances in the architecture.

Performance on Unseen Layout Strings. Now, we evaluate various approaches for their ability to detect snippet patterns that were not encountered by the approach during its training. This would test one-shot detection capabilities of the approaches. We distinguish seen-unseen classes by checking whether a layout string pattern in testset appeared in the trainset or not. The testset for Flamingo contains 1558 seen layout patterns and 353 unseen layout patterns; similarly, PubLayNet testset comprises of 17 seen and 6 unseen layout patterns. When inference is performed separately on the seen-unseen split, we obtain results as shown in table 4. The numbers depict MONOMER's superiority over other approaches in correctly identifying unseen layout strings, and thus, underscore its efficiency in inferring layout strings of even the unseen snippet patterns.

Model	*Flamingo*		*PubLayNet*	
	Seen	Unseen	Seen	Unseen
NCC	0.00	0.00	0.01	0.00
SSD	0.81	1.95	0.37	7.00
BHRL	47.50	42.30	16.10	16.00
LayoutLMv3	53.98	42.03	24.48	18.44
MONOMER	**71.33**	**57.82**	**31.86**	**31.27**

Table 4: Study of generalization capabilities of various approaches in one-shot setting (numbers in mAP).

6. Conclusion and Future Work

In this work, we propose a multi-modal one-shot detection setting for enhancing search within documents. Discussing the similarity in the context of documents, we propose a similarity criterion that allows generation of large amount of data required for testing out different approaches. Then, we propose a cross-attention based solution that is built upon insights into how various document modalities for queried snippet and target documents are inter-related. Our approach shows better performance compared to other approaches and its own single modality variants for the task of one-shot document snippet detection. In future, we wish to extend this work to other multi-modal content such as info-graphics, advertisement flyers, etc. which would further enhance search in document capabilities.

References

[1] OECD 2020. Digital transformation in the age of covid-19: Building resilience and bridging divides. *Digital Economy Outlook 2020 Supplement, OECD, Paris, www.oecd.org/digital/digital-economy-outlook-covid.pdf*, 2020.

[2] Milan Aggarwal, Hiresh Gupta, Mausoom Sarkar, and Balaji Krishnamurthy. Form2Seq : A framework for higher-order form structure extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3830–3840, Online, Nov. 2020. Association for Computational Linguistics.

[3] Milan Aggarwal, Mausoom Sarkar, Hiresh Gupta, and Balaji Krishnamurthy. Multi-modal association based grouping for form structure extraction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2075–2084, 2020.

[4] Alireza Alaei and Mathieu Delalandre. A complete logo detection/recognition system for document images. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 324–328, 2014.

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

[6] Michael W. Berry and Malú Castellanos. Survey of text mining: Clustering, classification, and retrieval. 2007.

[7] Galal M. Binmakhashen and Sabri A. Mahmoud. Document layout analysis: A comprehensive survey. *ACM Comput. Surv.*, 52(6), oct 2019.

[8] Glenn A Bowen. Document analysis as a qualitative research method. *Qualitative research journal*, 2009.

[9] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1), jan 2012.

[10] Ding-Jie Chen, He-Yen Hsieh, and Tyng-Luh Liu. Adaptive image transformer for one-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12247–12256, 2021.

[11] Jiaxin Cheng, Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Qatm: Quality-aware template matching for deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Adobe Document Cloud. Searching pdfs. 2021. publisher: Adobe.

[14] M.B. Hisham, Shahrul Nizam Yaakob, R.A.A Raof, A.B A. Nazren, and N.M. Wafi. Template matching using sum of squared difference and normalized cross correlation. In *2015 IEEE Student Conference on Research and Development (SCOReD)*, pages 100–104, 2015.

[15] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. *Advances in neural information processing systems*, 32, 2019.

[16] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. *arXiv preprint arXiv:2204.08387*, 2022.

[17] P Kiran, BD Parameshachari, J Yashwanth, and KN Bharath. Offline signature recognition using image processing techniques and back propagation neuron network system. *SN Computer Science*, 2(3):1–8, 2021.

[18] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018.

[19] Junlong Li, Yiheng Xu, Tengchao Lv, Lei Cui, Cha Zhang, and Furu Wei. Dit: Self-supervised pre-training for document image transformer, 2022.

[20] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5652–5660, 2021.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Keith Macdonald. Using documents 12. *Researching social life*, page 194, 2001.

[24] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[25] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Siamese network features for image matching. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 378–383. IEEE, 2016.

[26] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. *Levenshtein Distance: Information Theory, Computer Science, String (Computer Science), String Metric, Damerau?Levenshtein Distance, Spell Checker, Hamming Distance*. Alpha Press, 2009.

[27] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[28] Edouard Oyallon and Julien Rabin. An analysis of the surf method. *Image Processing On Line*, 5:176–218, 2015.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[30] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[32] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[33] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

[34] Mausoom Sarkar, Milan Aggarwal, Arneh Jain, Hiresh Gupta, and Balaji Krishnamurthy. Document structure extraction using prior based high resolution hierarchical semantic segmentation. In *European Conference on Computer Vision*, pages 649–666. Springer, 2020.

[35] Rubèn Tito, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. Icdar 2021 competition on document visual question answering. In *International Conference on Document Analysis and Recognition*, pages 635–649. Springer, 2021.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[37] Jian Wu, Zhiming Cui, Victor S Sheng, Pengpeng Zhao, Dongliang Su, and Shengrong Gong. A comparative study of sift and its variants. *Measurement science review*, 13(3), 2013.

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

[39] Hanqing Yang, Sijia Cai, Hualian Sheng, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, Yong Tang, and Yu Zhang. Balanced and hierarchical relation learning for one-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7591–7600, 2022.

[40] Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324, 2017.

[41] Jae-Chern Yoo and Tae Hee Han. Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28(6):819–843, 2009.

[42] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, page 103514, 2022.

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