LaTr: Layout-Aware Transformer for Scene-Text VQA

Ali Furkan Biten\textsuperscript{1}† Ron Litman\textsuperscript{2}† Yusheng Xie\textsuperscript{2} Srikar Appalaraju\textsuperscript{2} R. Manmatha\textsuperscript{2}

\textsuperscript{1}Computer Vision Center, UAB, Spain, \textsuperscript{2}AWS AI Labs
abiten@cvc.uab.es \{litmanr, yushx, srikara, manmatha\}@amazon.com

Abstract

We propose a novel multimodal architecture for Scene Text Visual Question Answering (STVQA), named Layout-Aware Transformer (LaTr). The task of STVQA requires models to reason over different modalities. Thus, we first investigate the impact of each modality, and reveal the importance of the language module, especially when enriched with layout information. Accounting for this, we propose a single objective pre-training scheme that requires only text and spatial cues. We show that applying this pre-training scheme on scanned documents has certain advantages over using natural images, despite the domain gap. Scanned documents are easy to procure, text-dense and have a variety of layouts, helping the model learn various spatial cues (e.g. left-of, below etc.) by tying together language and layout information. Compared to existing approaches, our method performs vocabulary-free decoding and, as shown, generalizes well beyond the training vocabulary. We further demonstrate that LaTr improves robustness towards OCR errors, a common reason for failure cases in STVQA. In addition, by leveraging a vision transformer, we eliminate the need for an external object detector. LaTr outperforms state-of-the-art STVQA methods on multiple datasets. In particular, +7.6\% on TextVQA, +10.8\% on ST-VQA and +4.0\% on OCR-VQA (all absolute accuracy numbers).

1. Introduction

Scene-Text VQA (STVQA) aims to answer questions by utilizing the scene text in the image. It requires reasoning over rich semantic information conveyed by various modalities – vision, language and scene text. Fig. 1 (a) depicts representative samples in STVQA, showcasing a model’s desired abilities, including: (1) a-priori information and world knowledge such as knowing what a website looks like (left image); and (2) the capability to use language, layout, and visual information (middle and right images).

In this work, we introduce Layout-Aware Transformer (LaTr), a multimodal encoder-decoder transformer based model for STVQA. We begin by exploring how far language and layout information can take us in STVQA. In Fig. 1 (b) we visualize the information extracted by the OCR system, showing that some questions only require text features, some require both text and layout information and only some need beyond that. Accounting for this, we propose a layout-aware pre-training and architecture.

Recently, Yang et al. \cite{74} demonstrated the advantages in pre-training STVQA models on natural images, proposing text-aware pre-training (TAP) scheme, which is designed to foster multi-modal collaboration. Acquiring large quantities of natural images with text is challenging and hard to scale, as most natural images do not contain scene text.
Even when they do, the amount of text is often sparse (previous statistics suggest a median of only 6 words per image [67, 74]). In addition, and more importantly, TAP did not account for the importance of aligning the layout information with the semantic representations when designing the pre-training objectives.

To counter these drawbacks, we propose layout-aware pre-training based on a single objective using only text and spatial cues as input. Our pre-training forces the model to learn a joint representation which accounts for the interactions between text and layout information, benefiting the downstream task of STVQA. Despite the domain gap, we find that pre-training on documents has certain advantages over natural images. Scanned documents contain more text compared to natural images, therefore it is easier to scale the experiment and expose the model to more data. Words in documents are usually complete sentences, helping the model better learn semantics beyond a simple bag of words. Moreover, scanned documents provide varied layouts, leading to effective alignment between language and spatial features. Lastly, performing pre-training without visual features reduces computational complexity substantially.

Our model utilizes a vision transformer [13] for extracting visual features, thus replacing the extensive need for an external object detector [21, 25, 74]. Moreover, in practice, current STVQA models exploit a dataset-specific vocabulary with a pointer mechanism for decoding [17, 21, 24, 25, 71, 74–76], creating an over-reliance on the fixed vocabulary and leaving no room for fixing OCR errors. Our model performs vocabulary-free decoding, does well even on answers out-of-vocabulary, and even overcomes OCR errors in some cases. LaTr outperforms the state-of-the-art STVQA methods by large margins on multiple public benchmarks. To summarize, the key contributions of our work are:

1. We recognize the key role language and layout play in STVQA and propose a layout-aware pre-training and architecture to account for that.
2. We pinpoint a new symbiosis between documents and STVQA via pre-training. We show empirically that documents are beneficial for tying together language and layout information despite the huge domain gap.
3. We show that existing methods perform poorly on out-of-vocabulary answers. LaTr does not require a vocabulary, does well even on answers that are not in the training vocabulary, and can even overcome OCR errors.
4. We provide extensive experimentation and show the effectiveness of our method by advancing the state-of-the-art by +7.6% on TextVQA and +10.8% on ST-VQA and +4.0% in OCR-VQA dataset.

2. Related Work

Pre-training and Language Models. The low cost of obtaining language text combined with the success of pre-training, language models [12, 40, 52, 53] has shown remarkable success in machine translation, natural language understanding, question answering and more. Recently, numerous studies [2, 10, 22, 28, 34–37, 42, 43, 61, 62, 77] showed the benefits of pre-training multi-modal architectures for vision and language tasks. Yang et al. [74] demonstrated, for the first time, the effectiveness of pre-training in scene text VQA by using masked language modeling and image-text matching as pretext tasks. In this paper, we show that tying together language and layout information via a simple layout-aware pre-training scheme is beneficial for scene text VQA. Moreover, we perform pre-training over scanned documents and discover that, despite the domain gap, documents can be leveraged for task of STVQA.

Vision-language tasks incorporating scene text. Recently, integrating reading into the vision and language tasks has become imperative, especially in VQA and captioning where the models were known to be illiterate [8, 58]. Since the usage of text can be quite distinct in terms of the environment, several papers introduce new datasets for various contexts in which text appears; ST-VQA [9], TextVQA [58] in natural images; OCR-VQA [49] in book and movie covers; DocVQA [47] in scanned documents; InfoVQA [46] in info-graphics. Moreover, STE-VQA [70] is proposed for multi-lingual VQA and TextCaps [57] for captioning on natural images. There are several papers published on scene text VQA. LoRRa [58] extended Pythia [23] with a pointer network [68] to select either from a fixed vocabulary or from OCR tokens. M4C [21] also used pointer networks but instead used multi-modal transformers [66] to encode all modalities together. SA-M4C [25] build on top of M4C by providing supervision on self-attention weights. MM-GNN [16] builds separate graphs for different modalities by utilizing graph neural networks [29]. Instead of having separate graphs for each modality, SMA [15] introduces a single graph that encodes all modalities. [78] proposes to use an attention mechanism to fuse pairwise modalities.

LaTr enriches the language modality with layout information via pre-training to achieve state-of-the-art performance across multiple benchmarks. Our model is generative in nature and as such alleviates the problem of vocabulary reliance current methods suffer from. In addition, we will show that LaTr is more robust to OCR errors, one of the most common reasons for failure cases in STVQA [21, 74].

3. Method

In this section, we describe in detail our model architecture and our pre-training strategy, as seen in Fig. 2. LaTr consists of three main building blocks. First, a language model pre-trained on only text. Second, use of spatial embedding for OCR tokens bounding box in conjunction with further layout-aware pre-training on documents, as depicted in Fig. 2 (a). Finally, a ViT architecture [13] for obtaining
As T5 was trained on just text data, we perform further pre-training to effectively align the layout information (in form of the 2-D spatial embedding) and the semantic representations. To the best of our knowledge, we are the first to propose pre-training on
documents instead of natural images for the task of scene text VQA. The motivation for selecting documents is that they are a source of rich text environment in a variety of complex layouts. Inspired by [53], we perform a layout-aware de-noising pre-training task, which includes the 2-D spatial embedding, as seen in Fig. 2 (a). This enables the use of weak data with no answer annotations in the pre-training stage. Like the normal de-noising task, our layout-aware de-noising task masks a span of tokens and forces the model to predict the masked spans. Unlike the normal de-noising task, we also give the model access to the rough location of the masked tokens, which encourages the model to fully utilize the layout information when completing this task.

More formally, let \( O = \{O_1, O_2, \ldots, O_n\} \) be the set of all OCR tokens (strings) and \( B = \{B_1, B_2, \ldots, B_n\} \) be the corresponding bounding box information, where \( B_j = (x'_0, y'_0, x'_1, y'_1, w^j, h^j) \). Now, let \( M_l = \{j, j + 1, \ldots, j + k\} \) be the \( th \) mask span where \( j \) is the starting index to mask such that \( \max(M_l) < \min(M_{l+1}) \). Then, \( \{O_j, \ldots, O_{j+k}\} \) and \( \{B_j, \ldots, B_{j+k}\} \) are replaced by \( \tilde{O}_l \) (a special indexed mask token) and \( \tilde{B}_l \) (the span’s minimal containing bounding box) in the following manner:

\[
\tilde{O}_l = \langle \text{extra_id}, \rangle, \text{where } l \in \{0, \ldots, k-1\} \\
\tilde{B}_l = (\min(\{x'_0\}), \min(\{y'_0\}), \max(\{x'_1\}), \max(\{y'_1\})) \\
\text{where } j \leq i \leq j + k
\]

(2)

where the height and width of the masked tokens’ bounding box are calculated with the coordinates of \( \tilde{B}_l \).

Essentially, we have replaced a span of words tokens \( \{O_j, \ldots, O_{j+k}\} \) and their corresponding bounding boxes \( \{B_j, \ldots, B_{j+k}\} \) with a special token \( \tilde{O}_l \) and a corresponding “loose” bounding box. In other words, when we mask the span of words, we select the minimum of the top-left coordinates and the maximum of the bottom-right ones. The reasons are twofold. First, we do not want our model to know precise token boxes because that would reveal how many tokens are masked. Second, we choose not to mask the bounding boxes completely because then the model does not know where the text should appear in the document and cannot use the correct spatial context effectively. So, we prevent the model from taking shortcuts, but at the same time give it enough information to learn. The masked token \( \tilde{O}_l \) and its bounding box \( \tilde{B}_l \) are then embedded using Eq. (1) like any other regular token. We use cross-entropy loss to predict all the masked tokens’ original text.

**Visual Features** Most previous methods utilized an external pre-trained object detector [21,74] for extracting objects labels, visual object features and visual OCR features. In this work, we diverge from the literature and leverage a Vision Transformer (ViT) [13]. The ViT is an image classification network which is pre-trained and fine-tuned on ImageNet [11]. We utilize ViT in our architecture only in the fine-tuning stage, and we freeze all the layers except the last fully connected projection layer we add. Formally, an image \( I \) having the dimension of \( H \times W \times C \) is reshaped into 2D patches of size \( N \times (p^2 \cdot C) \), where \((H,W)\) is the height and width, \(C\) is the number of channels, \((P, P)\) is the resolution of each image patch, and \( N = HW/P^2 \) is the final number of patches. As depicted in Fig. 2 (b), we utilize a linear projection layer to map the flattened patches to \( D \) dimensional space and feed them to the ViT. We pass the full ViT output (containing \([\text{class}]\) token) sequence to a trainable linear projection layer and then feed it to the transformer encoder. Position embeddings are added to the patch embeddings to retain positional information. We denote the final visual output as \( V = \{V_0, \ldots, V_N\} \).

LaTr So far, we explained the building blocks of our method, now we describe how we put it all together, as depicted in Fig. 2 (b). After pre-training the language modality of the model with layout information, we input all three modalities, namely: image, OCR information and question to the transformer encoder. Let \( V = \{V_0, \ldots, V_N\} \) be a set of visual patch features such that \( V_0 \) is the \([\text{class}]\) embedding, \( Q = \{W_1, \ldots, W_m\} \) be the question tokenized into \( W_i \) and \( O = \{O_1, O_2, \ldots, O_n\} \) be the OCR tokens. We embed the OCR tokens and questions using Eq. (1) to obtain encoded OCR tokens \( E \) and encoded question features \( E^q \). For the 2-D spatial embedding of each \( W_i \), we use fixed values \((x_0 = y_0 = 0, x_1 = y_1 = 1000)\). Finally, we concatenate all the inputs \([V; E; E^q]\) to feed to the multimodal transformer encoder-decoder architecture. Cross entropy loss is used to fine-tune our model.

4. Experiments

In this section, we experimentally examine our method, comparing its performance with state-of-the-art methods. We consider the standard benchmarks of TextVQA [58], ST-VQA\(^2\) [9] and OCR-VQA [49]. For pre-training we consider the same datasets used in [7,74] with the addition of the Industrial Document Library (IDL)\(^3\). The IDL is a collection of industry documents hosted by UCSF. It hosts millions of documents publicly disclosed from various industries like tobacco, drug, food etc. The data from the website amounts to about 13M documents, translating to about 64M pages of various document images. We further extracted OCR for each document using Textract OCR\(^4\). Implementation details and further information on all datasets

\(^2\)We use ST-VQA for denoting the dataset proposed in [9], and STVQA for denoting the general task of scene text VQA.

\(^3\)https://www.industrydocuments.ucsf.edu/

\(^4\)https://aws.amazon.com/textract/
In this work, we experiment with Amazon Text-in-Image to a more recent one than Rosetta and gradually increase the model capacity can get the best performance on both datasets. Finally, increasing our model capacity to LaTr-Large further boosts performance to 61.6% (+7.6% from [74]).

**ST-VQA Results** Tab. 2 presents the accuracy on ST-VQA [9] in the unconstrained setting. LaTr uses the Amazon-OCR and is pre-trained on IDL and fine-tuned on the training set of ST-VQA. LaTr-Large is also fine-tuned with TextVQA. The behaviour observed in TextVQA is consistent with ST-VQA dataset, LaTr-Large and LaTr-Base-Large outperforming the previous art [74] by +8.26% and +10.81%, respectively. Moreover, we show a similar trend on OCR-VQA [49] dataset where the discussion and the numbers can be found in Appendix E.

**Qualitative Analysis** In Fig. 4 we depict five different question categories which are representative of the capa-

<table>
<thead>
<tr>
<th>Method</th>
<th>OCR System</th>
<th>Pre-Training Data</th>
<th>Extra Finetune</th>
<th>No. of Param.</th>
<th>Val Acc.</th>
<th>Test Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4C [21]</td>
<td>Rosetta-en</td>
<td>X</td>
<td>X</td>
<td>200M</td>
<td>39.40</td>
<td>39.01</td>
</tr>
<tr>
<td>SMA [15]</td>
<td>Rosetta-en</td>
<td>X</td>
<td>-</td>
<td>40.05</td>
<td>40.66</td>
<td></td>
</tr>
<tr>
<td>CRN [39]</td>
<td>Rosetta-en</td>
<td>X</td>
<td>-</td>
<td>40.39</td>
<td>40.96</td>
<td></td>
</tr>
<tr>
<td>LaAP-Net [20]</td>
<td>Rosetta-en</td>
<td>X</td>
<td>-</td>
<td>40.68</td>
<td>40.54</td>
<td></td>
</tr>
<tr>
<td>TAP [74]</td>
<td>Rosetta-en</td>
<td>TextVQA</td>
<td>X</td>
<td>200M</td>
<td>-44.06</td>
<td>-</td>
</tr>
<tr>
<td>LaTr-Small</td>
<td>Rosetta-en</td>
<td>X</td>
<td>X</td>
<td>149M</td>
<td>41.84</td>
<td>-</td>
</tr>
<tr>
<td>LaTr-Base</td>
<td>Rosetta-en</td>
<td>X</td>
<td>X</td>
<td>311M</td>
<td>44.06</td>
<td>-</td>
</tr>
<tr>
<td>LaTr-Base</td>
<td>Rosetta-en</td>
<td>IDL</td>
<td>X</td>
<td>311M</td>
<td>48.38</td>
<td>-</td>
</tr>
<tr>
<td>SA-M4C [25]</td>
<td>Google-OCR</td>
<td>X</td>
<td>ST-VQA</td>
<td>200M</td>
<td>45.4</td>
<td>44.6</td>
</tr>
<tr>
<td>M4C [21, 74]</td>
<td>Microsoft-OCR</td>
<td>X</td>
<td>ST-VQA</td>
<td>200M</td>
<td>45.22</td>
<td>-</td>
</tr>
<tr>
<td>TAP [74]</td>
<td>Microsoft-OCR</td>
<td>TextVQA</td>
<td>X</td>
<td>200M</td>
<td>49.91</td>
<td>49.71</td>
</tr>
<tr>
<td>TAP [74]</td>
<td>Microsoft-OCR</td>
<td>TextVQA, ST-VQA</td>
<td>ST-VQA</td>
<td>200M</td>
<td>50.57</td>
<td>50.71</td>
</tr>
<tr>
<td>LOGOS [44]</td>
<td>Microsoft-OCR</td>
<td>X</td>
<td>ST-VQA</td>
<td>-</td>
<td>51.53</td>
<td>50.08</td>
</tr>
<tr>
<td>TAP [74]</td>
<td>Microsoft-OCR</td>
<td>TextVQA, ST-VQA, TextCaps, OCR-CC</td>
<td>ST-VQA</td>
<td>200M</td>
<td>54.71</td>
<td>53.97</td>
</tr>
<tr>
<td>M4C [21]</td>
<td>Amazon-OCR</td>
<td>X</td>
<td>X</td>
<td>200M</td>
<td>47.84</td>
<td>-</td>
</tr>
<tr>
<td>LaTr-Base</td>
<td>Amazon-OCR</td>
<td>X</td>
<td>X</td>
<td>311M</td>
<td>52.29</td>
<td>-</td>
</tr>
<tr>
<td>LaTr-Base</td>
<td>Amazon-OCR</td>
<td>IDL</td>
<td>X</td>
<td>311M</td>
<td>58.03</td>
<td>58.86</td>
</tr>
<tr>
<td>LaTr-Large</td>
<td>Amazon-OCR</td>
<td>IDL</td>
<td>ST-VQA</td>
<td>311M</td>
<td>59.53</td>
<td>59.55</td>
</tr>
<tr>
<td>LaTr-Large</td>
<td>Amazon-OCR</td>
<td>IDL</td>
<td>X</td>
<td>856M</td>
<td>59.76</td>
<td>59.24</td>
</tr>
<tr>
<td>LaTr-Large</td>
<td>Amazon-OCR</td>
<td>IDL</td>
<td>ST-VQA</td>
<td>856M</td>
<td>61.05</td>
<td>61.60</td>
</tr>
<tr>
<td>LaTr-Large</td>
<td>Amazon-OCR</td>
<td>IDL</td>
<td>ST-VQA</td>
<td>856M</td>
<td>61.05</td>
<td>61.60</td>
</tr>
</tbody>
</table>

Table 1. Results on the TextVQA dataset [58]. As commonly done, the top part of the table presents results in the constrained setting that only uses TextVQA for training and Rosetta for OCR detection, while the bottom part is the unconstrained settings. LaTr advances the state-of-the-art performance, specifically by +6.43% and +7.63% on validation and test, respectively.

can be found in Appendix A and B, respectively. We note that throughout the rest of the paper, ‡ refers to the models fine-tuned with both TextVQA and ST-VQA, at the same time. “-Small”, “-Base” and “-Large” model sizes refer to architectures that have 6+6, 12+12 and 24+24 layers in encoder and decoder, respectively. For convenience, we refer to LaTr-Base as LaTr.

**TextVQA Results** Similar to previous work [74], we define two evaluation settings. The former is the constrained setting that only uses TextVQA for training and Rosetta for OCR detection. The latter is the unconstrained setting, in which we present our best performance with the state-of-the-art. The first part of Tab. 1 reports the accuracy under the constrained setting. As can be appreciated, LaTr-Small outperforms M4C (+2.44%), with fewer parameters. Increasing the model capacity of LaTr results in a performance gain of +2.22% (additional discussion on the model capacity can be found in Appendix D). In addition, LaTr achieves the same performance as TAP [74] without any pre-training, demonstrating the effectiveness of our model. Furthermore, when LaTr is pre-trained on IDL, performance increase from 44.06% to 48.38% (+4.32%) using the Rosetta OCR. This clearly shows the effectiveness of layout-aware pre-training on scanned documents to the task of scene text VQA, even in the constrained setting.

In the bottom part of Tab. 1 we modify the OCR system to a more recent one than Rosetta and gradually add additional training datasets (unconstrained settings). In this work, we experiment with Amazon Text-in-Image (Amazon-OCR)\(^5\) [65]. As seen, when using Amazon-OCR our method outperforms the M4C baseline, improving performance from 47.84% to 52.29% (+4.45%). Furthermore, when enabling pre-training, LaTr outperforms the previous art [74] by large margins from 54.71% to 58.03% (+3.32%) on validation and from 53.97% to 58.86% (+4.89%) on the test. We note that for [74] there is a -0.74% decrease between validation and test while for LaTr we observe an increase of +0.83%, demonstrating better generalization. Another critical point is that LaTr can benefit more when ST-VQA dataset is added as an extra fine-tune data. We believe this point to be critical since we do not have to train separate models for TextVQA and ST-VQA but rather one model that can get the best performance on both dataset. Finally, increasing our model capacity to LaTr-Large further boosts performance to 61.6% (+7.6% from [74]).

\(^5\) [https://docs.aws.amazon.com/rekognition/index.html]
OCR errors (Fig. 4 (a)). Most state-of-the-art OCR systems for scene text [6,14,38,50] operate on a word-level, and thus are unable to utilize image-level context. Current STVQA methods depend on a pointer network for decoding, which means they are bound by the performance of the OCR system at hand. Contrary to that, LaTr leverages image-level context and jointly with its generative nature, is able to correct OCR errors. Next, scene text VQA models are required to have the ability to understand language together with world knowledge (Fig. 4 (b)(c)). Both requirements are met in LaTr thanks to its extensive pre-training.

As seen in Fig. 4 (d), answering questions often requires reasoning over the relative spatial positions of the text in the image. Over the years several methods aimed at developing spatially aware models were proposed [25,44]. However, most of those methods are complex, not easy to implement and eventually led to minimal performance improvements. LaTr is pre-trained on documents with layout information, which leads to a spatially aware model without any complex architectural changes. The last category we analyze is long answers (Fig. 4 (e)). In practice, the existing pointer network decoding mechanism is also limited in ability to produce long answers. Furthermore, when pre-training is done

### Table 2. Results on the ST-VQA Dataset [9].

<table>
<thead>
<tr>
<th>Method</th>
<th>Val Acc.</th>
<th>Val ANLS</th>
<th>Test ANLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4C [21]</td>
<td>38.05</td>
<td>0.472</td>
<td>0.462</td>
</tr>
<tr>
<td>SA-M4C [25]</td>
<td>42.23</td>
<td>0.512</td>
<td>0.504</td>
</tr>
<tr>
<td>SMA [15]</td>
<td>-</td>
<td>-</td>
<td>0.466</td>
</tr>
<tr>
<td>CRN [39]</td>
<td>-</td>
<td>-</td>
<td>0.483</td>
</tr>
<tr>
<td>LaAP-Net [20]</td>
<td>39.74</td>
<td>0.497</td>
<td>0.485</td>
</tr>
<tr>
<td>LOGOS [44]</td>
<td>48.63</td>
<td>0.581</td>
<td>0.579</td>
</tr>
<tr>
<td>TAP [74]</td>
<td>50.83</td>
<td>0.598</td>
<td>0.597</td>
</tr>
<tr>
<td>LaTr-Base</td>
<td>58.41</td>
<td>0.675</td>
<td>0.668</td>
</tr>
<tr>
<td>LaTr(^2)-Base</td>
<td>59.09</td>
<td>0.683</td>
<td>0.684</td>
</tr>
<tr>
<td>LaTr(^2)-Large</td>
<td>61.64</td>
<td>0.702</td>
<td>0.696</td>
</tr>
</tbody>
</table>

Table 3. Zero Shot Performance of T5 Language Model on TextVQA. In this setting, T5-Base is pre-trained on C4 and fine-tuned on SQuAD [54], a reading comprehension dataset. Showing that a “blind” pre-trained language model can get up to 25.45% on natural images as in [74], the model hardly encounters long sentences. LaTr does not rely on a pointer network and is pre-trained on documents, in which text appears in a variety of lengths.

We provide further qualitative analysis and comparisons to previous work [21] in Appendix F. In addition, we display failure cases of our method on the TextVQA dataset. The failure cases are mostly composed of OCR errors, compositionality of spatial reasoning and visual attributes.

### 5. Ablation Studies

In this section, we provide insightful experiments which we deem crucial for the STVQA task and its future development. We start off by showing the significance of language understanding in STVQA. Then, we show the effectiveness of language and layout information and discuss the biases existing in STVQA benchmarks. Next, we study the effect of pre-training as a function of dataset size and type. Finally, we showcase our model’s robustness towards vocabulary and OCR errors. All the numbers are obtained by using the TextVQA validation set.

#### Zero-shot Language Models on TextVQA

To quantify the importance of language understanding in STVQA, we devise a novel zero-shot setting where we use the T5 language model pre-trained on C4 and only fine-tuned on SQuAD [54], a reading comprehension dataset. Tab. 3 presents the performance of this setting while varying the OCR system. Interestingly, even without any visual fea-

![Figure 4. Why is STVQA hard? Current state-of-the-art methods struggle to acquire various abilities which are needed for scene text VQA. We depict five representative abilities; fixing OCR errors, language understating, world knowledge, understating complex layouts and the ability to produce long answers. Our model is able to correctly answer each one of these examples. We refer the reader to Figure 4.]
models are known to exploit dataset biases \cite{Anderson18}. Thus, we investigate if there are any biases in the data and if it is possible to categorize them.

### Dataset Bias or Task Definition?

To get a better sense of the biases in TextVQA, we start by training a model where only questions are given as input. As can be seen in Tab. 4, our model is able to achieve 11.18% in a task that requires reading and reasoning about the text without \textit{the text}. Next, we study the effect of the OCR system by dividing the information provided by it into text token transcription, reading order and 2-D positional information. Reading order is the order where OCR tokens are extracted from left to right and top to bottom with respect to line boxes or text blocks. Reading order is so intertwined with OCR systems that it is not thought of as a detached feature.

As shown in Tab. 4, adding OCR tokens without any reading order gives us 41.77% and a fixed reading order already gets us to 50.37%, showing the importance of reading order for given OCR tokens. The gain becomes marginal when adding the 2-D positional and visual information without pre-training, +0.85% and +1.09%, respectively. However, when performing \textit{layout-aware} pre-training on documents, obtaining alignment between the layout information and the semantic representations, LaTr’s performance increases significantly by +7.01% to 57.38%.

In other words, we can already achieve SOTA on a \textit{Visual} Question Answering task without any visual features (other than using the images for OCR extraction). Finally, adding visual features still \textit{marginally} increases performance by around +0.7%. Recently, \cite{hendrycks2020nacl} showed a similar phenomenon using the M4C \cite{m4c} architecture, where visual information only slightly contributed to the performance, validating that this is not specific to our technique.

Regarding the comparison of the different visual backbones, we train our model with visual features extracted either from FRCNN \cite{Ferstl18} or ViT \cite{dosovitskiy2020image}. We note that the performance difference is very marginal when only TextVQA is used in fine-tuning. However, when TextVQA and ST-VQA are used together, the model with FRCNN features perform worse than the model without any visual features while ViT increases performance by +0.61%, demonstrating that ViT features can scale better with more data.

At this point, we would like to take a step back and discuss STVQA as a task. As we see it, our analysis can be interpreted from two viewpoints. The first viewpoint is how STVQA is defined as a task. In particular, is the STVQA task defined such that all (or a majority of) questions should require reasoning over all modalities (including visual features)? Regardless of the answer, we present a second viewpoint, a dataset bias. To better explore the bias perspective, in Appendix G we visualize question-image pairs sorted by the information required to answer them. Clearly, generating questions from the final category (\textit{i.e.} questions which require reasoning over all modalities) is not an easy task. Furthermore, we quantitatively showed that at-least 60% of the questions do not fall under the final category, allowing the model to extensively exploit language priors and make educative guesses. Both viewpoints lead us to wonder are visual features even needed for STVQA? Or better yet, is vision an artifact in STVQA task? We believe that visual features are of importance for the task of STVQA, however current benchmarks do not reflect it, making it harder to evaluate how much V matters in STVQA.

## Table 4. LaTr Ablation Studies on TextVQA

We ablate LaTr - Base by varying the building blocks of our method, including pre-training, input types and fine-tuning data. V refers to ViT and F refers to FRCNN as visual backbone, \textit{random} means OCR tokens are provided but presented in a random reading order.

<table>
<thead>
<tr>
<th>Model</th>
<th>2-D</th>
<th>Pre-training</th>
<th>OCR</th>
<th>Visual</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>11.18</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>V</td>
<td>11.74</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
<td>random</td>
<td>X</td>
<td>41.77</td>
</tr>
<tr>
<td>LaTr</td>
<td>X</td>
<td>X</td>
<td>V</td>
<td></td>
<td>50.37</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>51.22</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>52.29</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>57.38</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>F</td>
<td>58.11</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>58.03</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>58.92</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>58.45</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>59.53</td>
</tr>
</tbody>
</table>

## Table 5. The Effect of Pre-training

Ablation studies on pre-training as a function of different datasets type and size.

<table>
<thead>
<tr>
<th>Model</th>
<th>Pre-training Data</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>50.37</td>
</tr>
<tr>
<td>LaTr</td>
<td>TextVQA</td>
<td>51.81</td>
</tr>
<tr>
<td>LaTr-Base</td>
<td>TextVQA,ST-VQA,TextCaps,OCR-CC</td>
<td>54.22</td>
</tr>
<tr>
<td></td>
<td>IDL - 1M</td>
<td>55.12</td>
</tr>
<tr>
<td></td>
<td>IDL - 11M</td>
<td>56.28</td>
</tr>
<tr>
<td></td>
<td>IDL - 64M</td>
<td>58.03</td>
</tr>
<tr>
<td></td>
<td>IDL -64M,TextVQA,ST-VQA,TextCaps,OCR-CC</td>
<td>58.51</td>
</tr>
<tr>
<td></td>
<td>IDL - 64M</td>
<td>59.53</td>
</tr>
<tr>
<td></td>
<td>IDL -64M,TextVQA,ST-VQA,TextCaps,OCR-CC</td>
<td>59.06</td>
</tr>
</tbody>
</table>
Vocabulary Reliance and Robustness Towards OCR Errors

Current state-of-the-art methods predict the answer through an amalgamation of a pointer mechanism and a dataset-specific 5K most frequent vocabulary. The usage of a vocabulary is limiting in a real-world scenario and may result in high performance on in-vocabulary answers but lead to poor performance on out-of-vocabulary ones, in other words, lack of generalization. This is clearly observed in Tab. 6 where M4C [21] exhibits a heavy reliance on the fixed vocabulary as the gap between categories is $-12.7\%$. Contrary to that, LaTr is not limited to any handcrafted dataset-specific vocabulary. Its gap between in and out of the training vocabulary is only $-1.58\%$.

Finally, we experimentally display that our model is more robust to OCR errors compared to M4C architecture.

<table>
<thead>
<tr>
<th>Model</th>
<th>All</th>
<th>InVoc.</th>
<th>OutVoc.</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>M4C [21]</td>
<td>47.84</td>
<td>51.07</td>
<td>38.37</td>
<td>12.7</td>
</tr>
<tr>
<td>LaTr</td>
<td>59.53</td>
<td>59.93</td>
<td>58.35</td>
<td>1.58</td>
</tr>
</tbody>
</table>

Table 6. Vocabulary Reliance. Accuracy gap between answers with words in and out of vocabulary used by [21, 25, 74]. InVoc. and OutVoc. stand for in and outside the vocabulary, respectively.

The Effect of Large-Scale Pre-Training Tab. 5 demonstrates the benefits of pre-training while varying the datasets type and scale. First, we explore the effect of pre-training on natural images with visual features (as done in [74]) using our architecture. In particular, we add the image-text matching objective and leverage the same datasets (which we term TAP-datasets) as in [74]. Pre-training only on TextVQA (Tab. 5), provides only $+1.5\%$ improvement for us compared to [74] reporting $+5\%$. The same behaviour of diminished gain is also observed with TAP-datasets.

Next, we compare IDL and TAP-datasets in pre-training. Even pre-training on 1M documents, LaTr’s performance increases by almost $+5\%$, which is more than the combination of all TAP-datasets. This is inspiring for two reasons, one of which is 1M documents are less than two thirds the size of TAP-datasets [74]. Secondly, our model is pre-trained with a simple de-noising objective and no visual features, making the pre-training significantly faster (around 23 times) compared to TAP [74] which is pre-trained with visual features, scene text features and multiple losses. We also argue that IDL is a better bed for layout-aware pre-training since it provides varied layouts to better align with language. Finally, we discuss the effect of increasing the size of IDL. Adding an order of magnitude more data only result in $+1\%$ or $+2\%$ increase. We emphasize that 64M documents hardly seems the saturation point for LaTr, i.e. more pre-training data can still improve the performance, especially when also increasing the model capacity.

Vocabulary Reliance and Robustness Towards OCR Errors

Current state-of-the-art methods predict the answer through an amalgamation of a pointer mechanism and a dataset-specific 5K most frequent vocabulary. The usage of a vocabulary is limiting in a real-world scenario and may result in high performance on in-vocabulary answers but lead to poor performance on out-of-vocabulary ones, in other words, lack of generalization. This is clearly observed in Tab. 6 where M4C [21] exhibits a heavy reliance on the fixed vocabulary as the gap between categories is $-12.7\%$. Contrary to that, LaTr is not limited to any handcrafted dataset-specific vocabulary. Its gap between in and out of the training vocabulary is only $-1.58\%$.

Finally, we experimentally display that our model is more robust to OCR errors compared to M4C architecture.
References


[34] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In AAAI, pages 11336–11344, 2020. 2


