Towards Models that Can See and Read

Roy Ganz*
Technion, Israel
ganz@cs.technion.ac.il

Oren Nuriel
AWS AI Labs
onurielamazon.com

Aviad Aberdam
AWS AI Labs
aaberdam@amazon.com

Yair Kittenplon
AWS AI Labs
yairk@amazon.com

Shai Mazor
AWS AI Labs
smazor@amazon.com

Ron Litman
AWS AI Labs
litmanr@amazon.com

Abstract

Visual Question Answering (VQA) and Image Captioning (CAP), which are among the most popular vision-language tasks, have analogous scene-text versions that require reasoning from the text in the image. Despite their obvious resemblance, the two are treated independently and, as we show, yield task-specific methods that can either see or read, but not both. In this work, we conduct an in-depth analysis of this phenomenon and propose UniTNT, a Unified Text-Non-Text approach, which grants existing multimodal architectures scene-text understanding capabilities. Specifically, we treat scene-text information as an additional modality, fusing it with any pretrained encoder-decoder-based architecture via designated modules. Thorough experiments reveal that UniTNT leads to the first single model that successfully handles both task types. Moreover, we show that scene-text understanding capabilities can boost vision-language models’ performance on general VQA and CAP by up to $2.69\%$ and $0.6\%$ CIDEr, respectively.

1. Introduction

In recent years, Vision-Language (VL) tasks, such as Visual Question Answering (VQA) [4, 18] and Image Captioning (CAP) [34, 2], have gained immense research interest [55, 39, 49, 30, 22, 46, 10, 47]. However, despite the remarkable success of VL models on these tasks, it was discovered a few years ago that such models are incapable of reasoning from the text in natural images [41, 8, 40]. This finding raised significant concerns, as understanding scented-text is crucial in almost any real-world application.

To address this issue, designated scene-text datasets were introduced for both VQA [41, 8] and CAP [40], aiming to highlight the importance of utilizing textual information in images. Following the introduction of the above datasets, a new line of research has arisen, focusing on scene-text-oriented tasks, evaluated individually and effectively dis-

*Work done during an Amazon internship.

Figure 1: See and read in VQA. Illustration of the possible three types of reasoning required in VQA image-question pairs and representative datasets distributions (middle). Samples from the ‘see’ (bottom left), ‘read’ (bottom right), and ‘see-∩-read’ (top) subsets are presented. Each sample includes an image, question, OCR, and model predictions.
that even the minority of works that evaluate both types of
tasks [46, 11, 3] do it on separate models, which are fine-
tuned per task, perpetuating the faulty tasks’ segregation.

Apart from being unjustified, this separation introduces
biases [7, 48], providing the models with prior knowledge
that implies which modality to focus on, which does not ex-
ist in real-world scenarios. Namely, it creates a shortcut that
encourages models to excel solely on a specific benchmark
by acquiring an understanding of either the visual or tex-
tual information in the image, but not both. In particular,
Biten et al. [7] recently showed that SOTA performance on
scene-text VQA can be achieved without using the visual
modality, and Wang et al. [48] revealed that existing scene-
text VQA models’ success stems from exploiting language
priors. Our combined evaluation effectively addresses this
problem by testing whether models can reason from both
types of information, as exploiting such data biases and pri-
ors would yield low combined results.

From a more high-level view, three categories span the
space of VL data; the first are examples that require reason-
ning over vision only (dominant in VQA [18] and CAP [9]),
the second are instances in which using scene-text informa-
tion solely is sufficient (dominant in scene-text VQA [8, 41]
and scene-text CAP [40]), and the third are ones in which
both are essential. We denote the three subsets as ‘see-
’read’, and ‘see∩-read’, respectively. For completion, the
whole space is denoted as ‘see∩-read’, the union of all
others. We illustrate this conceptual data distribution for
VQA in Fig. 1. Examining the performance of existing
VQA approaches over the three types of questions men-
tioned above, shown in Fig. 2, reveals that while some of
the methods [30, 31, 47] perform well on the first subset and
some [21, 52] on the second, none are optimal on the entire
domain. Moreover, throughout our analysis, we reveal that
the ‘see∩-read’ subset, in which both visual and textual in-
formation are needed for answering, is very challenging and
underrepresented, requiring a new dedicated benchmark.

In this work, while striving towards models that excel
on the entire space of VL data, we propose UniTNT, a
Unified Text-Non-Text model, which provides VL archi-
tectures with scene-text understanding capabilities. Spe-
cifically, we treat textual information in the image, i.e. tokens
and positions, as a third modality and introduce it into the
pretrained model. Adding a new modality to an already-
trained model is challenging and might lead to suboptimal
results [16, 43, 54, 3]. To overcome this, we encode such
information using a designated encoder and inject it into the
existing pretrained decoder via a novel fusing mecha-
nism that gradually shifts between VL features to textual-
enriched ones. Moreover, we propose scene-text-related
intermediate supervision to encourage the already-trained
model to leverage the newly added information. Being both
task and model agnostic by its design, our method can be
applied to any VL encoder-decoder-based architecture.

We evaluate UniTNT on both general and scene-text
benchmarks of VQA and CAP using combined evaluation
and show that it leads to the first single model perform-
ning well on both tasks. We show that our method can
be easily integrated into existing VL models, improving
their scene-text understanding substantially by applying it
to BLIP [30] and ALBEF [31]. Interestingly, such rea-
soning abilities boost the base model’s VQA results (e.g.,
improves BLIP [30] by 2.69% on VQAv2 [18]), while
achieving state-of-the-art competitive results on scene-text
VQA benchmarks. A similar trend exists in captioning,
where UniTNT enhances BLIP’s performance by 0.6 CIDEr
points on COCO Captions [9] while substantially boosting
its scene-text CAP performance. These improvements high-
light the significance of scene-text comprehension in VL
tasks, laying the foundation for future research on general
multimodal architectures that can leverage scene-text.

To summarize:

• We thoroughly analyze current methods and reveal that
  the faulty text-non-text task separation leads to models
  that either reason from visual or textual information in
  images, but not both.

• We introduce UniTNT, a model-agnostic method to grant
  reading capabilities to pretrained VL models by fusing
  the scene-text information as an additional modality.

• Extensive experiments show that our method not only im-
  proves the scene-text benchmarks’ results but also signif-
  icantly enhances the performance of VQA and CAP.
2. See and Read: Analyzing Methods and Data

In this paper, contrary to the common practice in VL research, we highlight the importance of models to “see” and “read” altogether and start by comprehensively analyzing such capability via a “see∪read”-oriented combined evaluation. Our analysis reveals that existing models’ reasoning abilities over both types of information are lacking, prompting the question of whether this limitation is due to inherent method constraints or biased data. Our evaluation focuses on the performance of leading general and scene-text-oriented models on VQA-v2 [18], TextVQA [41], and ST-VQA [8] for VQA, and COCO Captions [9] and TextCaps [40] for captioning.

2.1. Visual Question Answering

General VQA Methods: During the vision-language revolution, numerous methods [33, 39, 31, 30, 47, 49, 46, 3, 10, 55, 53, 13, 22] have been proposed for various multimodal tasks, including VQA, which have advanced the state-of-the-art. These methods can leverage vast online image-caption pairs via vision-language pretraining [32, 12, 39], followed by task-specific fine-tuning. However, a few years ago, such models were shown to be ineffective in reasoning from textual information in the scene, as they primarily focus on the images’ visual content [41, 8].

Nevertheless, such models have advanced significantly in the past few years. Thus, to reveal the current status of such models in scene-text understanding, we examine the performance of three leading VQA models, ALBEF [31], BLIP [30], and OFA [47], using unconstrained open-vocabulary generation, on scene-text VQA tasks. As seen in Tab. 1, although such methods perform well on VQA, as expected, their results on the analogous scene-text VQA datasets are unsatisfactory, testifying their incompetence in scene-text understanding. Interestingly, their inability to utilize scene-text information hinders its performance even on VQA, as we later show in Sec. 4.

Scene-Text VQA Methods: Several methods have been proposed to improve the scene-text understanding of VQA models [21, 19, 17, 52, 23, 36, 7]. These models utilize an off-the-shelf OCR system’s output alongside the image and question as input to a multimodal transformer. However, some recent studies [48, 7] have indicated that scene-text VQA datasets may have biases discouraging models from relying on the visual modality. To properly test such claims, we evaluate M4C [21] and TAP [52] on general VQA, which requires strong visual understanding and report the results in Tab. 1. As can be seen, M4C and TAP obtain only 27.70% and 18.81%, respectively. When compared to, for example, BLIP’s 76.59%, it testifies that, indeed, such methods disregard the visual information. Interestingly, although TAP consistently outperforms M4C on the scene-text benchmarks, it achieves lower results on the general one, implying the data biases in the former datasets.

2.2. Image Captioning

Similar to VQA’s analysis, we conduct a captioning combined evaluation using TextCaps and COCO Captions for both types of models and report the average CIDEr scores. Our empirical results in Tab. 1 demonstrate that while general models (BLIP and OFA) and scene-text ones (M4C-Captioner and TAP) perform well on their designated benchmarks, they fail to obtain satisfactory results on the analogous one. In particular, BLIP obtains a CIDEr score of only 61.9 on TextCaps, compared to 90.1 of M4C-Captioner. On the other hand, the latter achieves 4.7 on COCO captions, compared to BLIP’s 133.3. In addition, like in VQA, while TAP outperforms M4C in TextCaps, it does not occur on COCO Captions. These findings suggest that existing methods exhibit unsatisfactory performance when evaluated on both captioning benchmarks.

2.3. The Role of the Datasets in ‘See and Read’

We now examine whether this limitation stems from a lack of representative training data rather than method limitations. Specifically, we test if the inferior performance of
Figure 3: **An overview of UniTNT.** Our method endows existing general VL models with scene-text understanding capability. The OCR information is encoded separately and injected into the decoder via a gated cross-attention-based fusing mechanism as complemental information. \( L_{\text{OCR-BC}} \) and \( L_{\text{OCR-LM}} \) are auxiliary losses, enforcing the model to utilize the scene-text information. UniTNT newly introduced components are presented in bold. ‘See’, ‘Read’, and ‘Fusing’ related modules are in blue, orange, and red, respectively.

scene-text-oriented models on visual tasks and vice versa is solely due to the training data’s bias towards reasoning over solely one type of information. To test this claim, we merge two datasets, conduct combined training for both general and scene-text-oriented methods, and report the results in Tab. 1. As can be seen, while unified training leads to improved performance on both types of VL benchmarks, there is a substantial performance gap – scene-text models lag behind general ones on the general benchmarks and vice-versa. Nevertheless, these results indicate that reasoning from text and vision are not at odds and suggest a symbiotic relationship between the two tasks. Furthermore, they provide further motivation for avoiding the common practice of separating the tasks, as done in previous work [3, 11, 46]. To conclude, while joint training is a step forward, it is not enough to achieve our ultimate goal.

### 3. Method

In this section, we describe UniTNT, a method aimed to obtain our titular goal by granting pretrained general VL models the ability to reason over scene-text information during finetuning while retaining their original reasoning capabilities, depicted in Fig. 3. By doing so, we propose a change of perspective compared to top-performing ST methods, such as [7, 21, 52], that harnesses an OCR-oriented pretrained model but fails to enrich it with visual understanding during finetuning. Adapting pretrained models to consider additional inputs, absent during pretraining, is a non-trivial task tackled by recent literature [43, 3]. On the one hand, we wish to encourage the model to utilize the new stream of information and, on the other hand, to prevent it from neglecting the original stream. To address this, we encode the OCR information via a designated OCR encoder and fuse it residually, retaining the former stream of information and gradually shifting towards an OCR-enriched one. Moreover, we propose auxiliary losses, encouraging the pretrained decoder to utilize this information. Similarly to previous works [21, 52, 36, 7], we utilize an off-the-shelf-OCR system to extract the scene-text information.

### 3.1. Architecture

We design our architecture in a task-agnostic way – enabling compatibility with both visual question answering and image captioning tasks. In addition, UniTNT is model agnostic and can be applied to any encoder-decoder-based VL model. In this work, we integrate our approach into two top-performing open-source methods – ALBEF [31], and BLIP [30] as a case study, denoted as UniTNT\_ALBEF / BLIP.

**OCR Encoder** Rather than utilizing the pre-existing encoder to process the OCR alongside the visual modality, as in [7, 21, 52], we introduce a dedicated OCR encoder, which maps the scene-text information into features fed into the existing system’s decoder. This encoder receives the question alongside OCR information, namely tokens and 2-dimensional (2D) positional information, both extracted by the OCR system. The positional information was proven to be valuable for documents and scene-text understanding.
tasks [50, 51, 5, 7]. Not only that our approach outperforms the one that utilizes the pre-existing text encoder to process the OCR tokens (demonstrated in Sec. 5), but it also provides flexibility to address tasks that do not utilize a text encoder, such as image captioning.

Formally, each OCR instance is represented by \((t, x_0, y_0, x_1, y_1, w, h)\), namely, its word token, bounding box’s top-left, bottom-right, width, and height values, respectively. We embed each value separately using designated embedding layers \(E(i.e., \text{torch.nn.Embedding})\). Next, we sum the 2D representations, pass them via a 2-layer MLP and add it to the token’s representation, yielding the OCR representation:

\[
e_{\text{OCR}} = E_{\text{OCR}}(t) + \alpha \ast \text{MLP}(E_x(x_0) + E_y(y_0) + E_x(x_1) + E_y(y_1) + E_w(w) + E_h(h))
\]

where \(\alpha\) is a predefined hyperparameter. As for the question, we embed its tokens using the same embedding layer. Since both the OCR and the question representations are fed into the same model, we equip the question representations with pseudo-2D information corresponding to the size of the entire image, yielding the final question representation \(e_q\). Finally, we concatenate them to obtain the OCR encoder’s input, \(\{e_q^1, \ldots, e_q^M, e_{\text{OCR}}^1, \ldots, e_{\text{OCR}}^N\}\), where \(M\) and \(N\) are the lengths of the question and OCR, respectively.

**VL-OCR Decoder** To integrate the OCR information into the decoder, we add a dedicated OCR Cross Attention (CA) and a fusing mechanism, as visualized in Fig. 3. We place the OCR CA block parallel to the pre-existing VL CA module to enrich the decoded features with textual information in the image. This architectural design yields two data streams (visual and scene-text-oriented ones) that need to be merged adequately into a single VL-OCR representation. To this end, we introduce a fusing mechanism composed of a gated cross-attention mechanism, which gradually shifts from VL features to fused, OCR-enriched ones.

Formally, our fusing mechanism merges the output of our new OCR CA with the one of the VL CA, denoted as \(F_{\text{OCR}}\) and \(F_{\text{VL}}\), respectively. Specifically, this module receives two features sequences, \(F_{\text{OCR}}, F_{\text{VL}} \in \mathbb{R}^{B \times L \times C}\), and outputs \(F_{\text{fused}} \in \mathbb{R}^{B \times L \times C}\), where \(B, L, C\) are the batch size, sequence length and the number of channels, respectively. First, we concatenate \(F_{\text{OCR}}\) and \(F_{\text{VL}}\) across the channel dimension and insert them into a simple 2-layer MLP to obtain an attention map \(F_{\text{attn}} \in \mathbb{R}^{B \times L \times C}\). Next, we pass the element-wise product of \(F_{\text{OCR}}\) and \(F_{\text{attn}}\) in a tanh gating mechanism [20, 3]. The goal of the tanh gating is to enable gradual OCR blending with the VL one by multiplying its outputs with \(\text{tanh}(\beta)\), where \(\beta\) is a learnable parameter initialized to zero. At initialization, it ensures that the added modules are skipped, preserving the model pretraining’s data flow. Finally, we sum the output of the tanh gating with \(F_{\text{VL}}\) to obtain the fused features:

\[
F_{\text{attn}} = \text{MLP}(\text{concat}(F_{\text{VL}}, F_{\text{OCR}})),
\]

\[
F_{\text{fused}} = F_{\text{VL}} + \text{tanh}(\beta)(F_{\text{OCR}} \odot F_{\text{attn}}),
\]

where \(\odot\) is the Hadamard product.

### 3.2. Scene-text Auxiliary Losses

We propose two auxiliary losses, encouraging the model to utilize the scene-text signal rather than ignoring it - OCR Causal Language Modeling (OCR-LM) and OCR Binary Classification (OCR-BC).

**OCR Causal Language Modeling** To better fuse the scene-text information, we add a causal language modeling supervision over the OCR tokens. Specifically, we prepend the shifted OCR tokens (according to the OCR system reading order) to the inputs of the decoder and train the system to predict the next OCR token based on previous ones:

\[
L_{\text{OCR-LM}} = - \sum_{i=1}^{N} \log \left( P \left( t^i | t^{<i} \right) \right)
\]

where \(t^i\) is the \(i\)th OCR token. Minimizing such loss enforces the system to account for the scene-text signal, as desired. While variants of such a loss were previously used during pretraining [52, 7], we are the first to utilize it during finetuning. Moreover, inserting the OCR into the decoder at inference has another significant advantage, as it serves as a prefix and enables the model to condition its answers on the OCR. Such behavior is desirable since the OCR can provide meaningful information for general and scene-text VL tasks, as we experimentally demonstrate in Sec. 4.1.

**OCR Binary Classification** To obtain more meaningful and task-beneficial OCR encodings, we propose a binary classification objective of predicting whether each OCR token is a part of the ground-truth answer. We build a binary linear classifier on top of the outputs of the OCR encoder and train it using a binary cross-entropy loss. More specifically, since most of the OCR tokens are not part of the answer, we employ a weighted version, as such classification task is highly imbalanced. We denote this loss as \(L_{\text{OCR-BC}}\).

### 3.3. Training Procedure

So far, we have described the main building blocks in our method, and now, as illustrated in Fig. 3, we put it all together. First, we harness a trained general encoder-decoder VL model and modify it as described above in Sec. 3.1. Next, we freeze the VL model’s pre-existing image encoder, similarly to [54, 3], and train UniTNT on a unified dataset (i.e., general and scene-text VQA datasets.
Table 2: VQA results. Accuracy of general, scene-text oriented VQA methods and UniTNT using three training regimes – separate VQA and TextVQA and combined training, where non-open vocabulary methods results are in gray. ∆ indicates improvement over the base architecture in the same regime. These results highlight our method’s effectiveness, significantly improving the general VQA results by enriching VL models with scene-text understanding.

<table>
<thead>
<tr>
<th>Method</th>
<th>OCR System</th>
<th>VQA test-dev</th>
<th>VQA test-std</th>
<th>TextVQA val</th>
<th>TextVQA test</th>
<th>ST-VQA test</th>
<th>UniTNT test-ANLS</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimVLMbase</td>
<td>✓</td>
<td>79.32 79.56</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ALBEF</td>
<td>✓</td>
<td>75.51 -</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TAP</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LOGOS</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M4C</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M4C-Captioner</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>TAP</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LOGOS</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M4C</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M4C-Captioner</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Comb</td>
<td>✓</td>
<td>37.47 37.70</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

4. Experiments

In this section, we experimentally examine UniTNT, comparing its performance with state-of-the-art methods with a similar capacity on both VQA and CAP tasks, using separate and combined training. In particular, to better study the effects of our method, we test it and the baselines in three distinct training regimes: (i) separate training on the general datasets, (ii) separate training on the scene-text ones, and (iii) combined training approach, denoted as Comb. As we focus on models’ see and read capabilities, we emphasize the combined training regime and view it as the most crucial one. However, the separate training regimes can provide insights into the impact of scene-text understanding on the general benchmarks and the biases within the scene-text datasets. As in Sec. 2, for each of the regimes, we consider three standard benchmarks for VQA: VQA v2 [18], TextVQA [41] and ST-VQA [8], and two for CAP: COCO Captions [9] and TextCaps [40]. We report the performance on each benchmark and the non-weighted averaged one (combined evaluation) to quantify the models’ reasoning capabilities from both visual and textual information as a single number. For VQA, we calculate this score only on VQA v2 and TextVQA test sets. Lastly, in Sec. 4.3, we present a new subset evaluation setting for scene-text VQA to measure the model’s ability to answer questions requiring reasoning over all modalities simultaneously. For all datasets, we extract OCR information using Amazon Text-in-Image[1] [35, 38, 1, 26]. The supplementary materials list the implementation details, additional dataset information and for completeness, a comparison with other methods, disregarding the models’ size.

4.1. Visual Question Answering Experiments

We integrate our approach to two models, ALBEF and BLIP, denoted as UniTNT_ALBEF and UniTNT_BLIP, respectively, and report their performance using three training regimes: (i) VQA, (ii) TextVQA, and (iii) Comb., as shown in Tab. 2. In the first regime, training UniTNT_BLIP exclusively on VQA v2 results in performance improvements of +3.19% and +12.16% on VQA, and TextVQA, respectively, leading to a significant boost of +7.67% in the average score. Even though VQA v2 mainly focuses on reasoning from visual information, these results stress the importance of scene-text understanding in this benchmark and the effectiveness of our method. Interestingly, despite the marginal presence of OCR in VQA v2, UniTNT_BLIP manages to effectively harness it and obtain 35.90% on

Table 3: CAP results. CIDEr scores of general, scene-text oriented CAP methods and UniTNT using three training regimes – separate Caps and TextCaps and combined training. ∆ indicates improvement over the base architecture in the same regime. These results highlight our method’s effectiveness, significantly improving the general CAP results by enriching VL models with scene-text understanding.

<table>
<thead>
<tr>
<th>Method</th>
<th>OCR System</th>
<th>COCO Karpathy-test</th>
<th>TextCaps Karpathy-test</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>VinVL</td>
<td>✓</td>
<td>129.3 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LEMONbase</td>
<td>✓</td>
<td>133.3 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GITbase</td>
<td>✓</td>
<td>138.5 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SimVLMbase</td>
<td>✓</td>
<td>142.6 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OFAbase</td>
<td>✓</td>
<td>150.7 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BLP</td>
<td>✓</td>
<td>133.7 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>M4C-Captioner</td>
<td>✓</td>
<td>146.4 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Comb</td>
<td>✓</td>
<td>146.4 ± 23.7</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

https://docs.aws.amazon.com/rekognition/latest/dg/text-detection.html
We curate a subset out of TextVQA [41] validation set, containing only the samples which require reasoning over both vision and scene-text in the same question. Presented are representative examples from this subset, each includes an image, question, OCR input tokens, and model predictions. Green and red stand for correct and wrong predictions, respectively.

Table 4: TextVQA splits. Accuracy of leading scene-text VQA methods on the two non-overlapping subsets of TextVQA validation data, and the gap between them. ‘See∩-Read’ refers to our subset, in which reasoning over all modalities is needed for each sample. ‘Read’ stands for the rest of the TextVQA validation set.

TextVQA, outperforming BLIP that trained solely on TextVQA itself (27.72%). In the scene-text configuration, performance improves by +19.67% on TextVQA; however, it decreases by −3.15% on VQA. This reinforces previous findings [48, 7], suggesting that scene-text VQA datasets contain biases encouraging models to over-rely on the OCR and disregard the visual information. As BLIP’s scene-text understanding is very restricted, it cannot fully exploit such biases and retains its visual understanding better, expressed via better VQAv2 results. The results indicate that despite the biases in scene-text VQA datasets, UniTNT can harness them without sacrificing the visual reasoning capability.

4.3. A subset for Reasoning Over All Modalities

As illustrated in Fig. 1, VQA data is composed of three categories. Some questions can be answered using just vision (‘see’), some by reasoning over the scene-text information only (‘read’), and some require reasoning over both modalities at once (‘see∩-read’). Since most of the questions in current benchmarks fall either in the ‘see’ or
'read' category, unifying them is beneficial for testing methods’ performance on the whole space, denoted by 'see-U-read', eliminating the model’s prior on whether a question is of type 'see' or 'read'. However, the more challenging and intriguing questions are the ones that require reasoning over scene-text and visual information altogether, denoted as 'see-∩-read'. To provide a more reliable way to evaluate VQA models on this questions’ category, we manually curate all such image-question pairs from the TextVQA [41] validation set, producing an evaluation subset of 480 image-question pairs out of the total 5000 (±10%).

This subset can serve as a foundation for measuring models’ capabilities on what we believe are the more challenging questions that the research community should tackle. In Fig. 4, we depict examples from this subset alongside the prediction of M4C [21], BLIP [30], and UniTNT. This qualitative analysis confirms that both scene-text-oriented methods and UniTNT on the non-overlapping subsets of TextVQA validation set, i.e., the 'TextVQA_{see∩read}' subset and its complementary set, 'TextVQA_{read}', expose the performance degradation that occurred on the former. As these findings suggest, our method leads to the best performance, affirming that it is indeed better at reasoning on scene-text and visual information simultaneously. Nevertheless, as can be seen, while UniTNT is a step forward, there is still a big room to improve on these types of challenging questions.

5. Ablation Studies

In this section, we study the effect of our key contributions and test the impact of freezing the vision encoder.

**Design Choices:** We ablate UniTNT’s components on both the general and scene-text-oriented datasets in Tab. 5, where all numbers are reported under the Comb. settings. Since the trends in CAP results are similar to VQA, we will focus on analyzing the latter. First, we report the added performance of a naive approach – simply inserting the OCR tokens as an additional input to BLIP’s existing text encoder, similar to [21, 52, 7]. As seen in Tab. 5, the accuracy on TextVQA improves by +10.59% (from 32.43% to 43.02%) while improving VQA results by +0.22%. Our designated OCR encoder increases TextVQA performance by 46.13% (+3.11%) while obtaining an additional +0.75% gain in VQA. Introducing our VL-OCR decoding scheme (denoted as “Fuse”) boosts us to 47.37% on TextVQA and an extra +0.24% on VQA. Furthermore, using \( L_{OCR-LM} \) significantly improves TextVQA performance by +5.22% (from 47.38% to 52.66%) while gaining an extra +0.21% on VQA. Finally, the combination of \( L_{OCR-BC} \) with the 2-D information gets us to 55.21% and 79.9% on TextVQA and VQA. Overall, UniTNT leads to significant +22.78% and +2.50% improvements on TextVQA and VQA over the combined trained BLIP.

**The Effect of Freezing the Visual Encoder:** Recently, a few works [43, 3, 54] have examined different freezing configurations to avoid knowledge forgetfulness when combining pretrained models. Inspired by these works, we examine the effect of freezing the Visual Encoder (VE) weights while applying UniTNT, preserving its valuable knowledge acquired in pretraining, and summarize the results in Tab. 6. As our findings suggest, freezing the VE significantly improves the results on VQA for both UniTNT_{ALBEF} and UniTNT_{BLIP} by +2.05% and +1.13%, and on TextVQA by +3.13% and +2.75%, respectively.

### Table 5: UniTNT design choices

<table>
<thead>
<tr>
<th>Method</th>
<th>Freeze VE</th>
<th>VQA test-dev</th>
<th>TextVQA val</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UniTNT_{ALBEF}</td>
<td>✓</td>
<td>77.40</td>
<td>43.02</td>
<td>109.7</td>
</tr>
<tr>
<td>UniTNT_{BLIP}</td>
<td>✓</td>
<td>79.86</td>
<td>52.66</td>
<td>113.8</td>
</tr>
<tr>
<td>UniTNT_{ALBEF}</td>
<td>✓</td>
<td>75.84</td>
<td>43.13</td>
<td>110.4</td>
</tr>
<tr>
<td>UniTNT_{BLIP}</td>
<td>✓</td>
<td>79.86</td>
<td>52.66</td>
<td>113.9</td>
</tr>
<tr>
<td>UniTNT_{ALBEF}</td>
<td>✓</td>
<td>78.65</td>
<td>47.38</td>
<td>118.3</td>
</tr>
<tr>
<td>UniTNT_{BLIP}</td>
<td>✓</td>
<td>79.86</td>
<td>52.66</td>
<td>113.8</td>
</tr>
<tr>
<td>UniTNT_{ALBEF}</td>
<td>✓</td>
<td>77.66</td>
<td>43.02</td>
<td>109.7</td>
</tr>
<tr>
<td>UniTNT_{BLIP}</td>
<td>✓</td>
<td>79.86</td>
<td>52.66</td>
<td>113.9</td>
</tr>
<tr>
<td>UniTNT_{ALBEF}</td>
<td>✓</td>
<td>77.66</td>
<td>43.02</td>
<td>109.7</td>
</tr>
<tr>
<td>UniTNT_{BLIP}</td>
<td>✓</td>
<td>79.86</td>
<td>52.66</td>
<td>113.9</td>
</tr>
</tbody>
</table>

### Table 6: Visual encoder freezing. VQA accuracy of UniTNT_{BLIP} and UniTNT_{ALBEF}, with and without freezing the visual encoder, attesting to the freezing’s importance.

6. Discussion and Conclusions

We wish to convey a few take-home messages to the VL research community. First, current SOTA methods cannot adequately reason over both scene-text and vision information. Our experiments demonstrate that this occurs even when combining training datasets, suggesting a fundamental limitation of existing methods. Second, our findings discover the symbiotic nature of these two types of reasoning capabilities, as performance on both tasks can be improved jointly. Moreover, by proposing UniTNT, we present the first single model that successfully handles both task types. Finally, we argue that the VL research community should strive to develop models that can simultaneously reason over vision, language, and scene-text. To facilitate this, we curate a suitable subset to serve as a benchmark foundation.
References


[41] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8317–8326, 2019. 1, 2, 3, 6, 7, 8, 12


A. Implementation Details

In this section, we provide full implementation specifics of UniTNT and divide it into three parts – (1) architecture; (2) training procedure; (3) Scene-text information.

A.1. Architecture

We harness the model agnosticism of UniTNT and apply it to two top-performing VL models. Specifically, we utilize the publicly-available code bases of ALBEF [31] and BLIP [30] and apply our method to them. We design our approach in a modular way enabling simple integration into existing models. Below we list the architectural specifics for both UniTNT_{ALBEF} and UniTNT_{BLIP}.

OCR Encoder We use a pretrained BERT-base4 as our encoder and introduce it with 2-dimensional information, as can be seen in Equation 1. Specifically, we use three separate embedding layers (i.e., torch.nn.Embedding) for the word token and its x and y axes positions for both the OCR and the question. In particular, we define the minimal and the maximal spatial position as 0 and 1000, respectively, and set these values for the question tokens (referred to as “pseudo-2D information” in the main paper). We restrict the number of OCR and question token lengths to 128 and 35, respectively. Next, we sum the 2D-related embeddings and pass them in a 2-layer MLP with a hidden dimension of 768 for additional processing. Finally, we multiply it by $\alpha$ (set to 0.1) and sum it with the token representation to obtain the final one fed into the encoder.

VL-OCR Decoder In order to introduce the pretrained decoder with scene-text information, we create new OCR Cross Attention (OCR-CA) blocks and place them in parallel to the existing VL ones. Such newly added components are identical to the existing ones and initialized with the pretrained weights of the latters’. To fuse the outputs of the OCR CA and the VL CA, $F_{OCR}$ and $F_{VL}$, we concatenate them along the channel dimension and pass them via attention based 2-layers MLP with a hidden size of 768 to obtain $F_{attn}$, an attention map that multiplies $F_{OCR}$ ($F_{OCR} \odot F_{attn}$). Namely, this mechanism highlights the important and meaningful features in $F_{OCR}$ and masks the less relevant ones. Then, we pass the multiplication output via a learnable gating module (by multiplying it by $\text{tanh}(\beta)$, where $\beta$ is learnable and initialized to 0), aimed to gradually blend the OCR features into the existing VL one.

Figure 5: OCR prevalence in VQAv2. Histogram of the number of OCR instances per-image in VQAv2 dataset.

A.2. Training Procedure

We train all of our models to minimize $\mathcal{L}_{\text{UniTNT}} = \mathcal{L}_{\text{base}} + \alpha_1 \mathcal{L}_{\text{OCR-LM}} + \alpha_2 \mathcal{L}_{\text{OCR-BC}}$ using 8 A100 GPUs, where $\alpha_1$ and $\alpha_2$ are hyperparameters.

Visual Question Answering We train both UniTNT_{ALBEF} and UniTNT_{BLIP} on a unified Text-Non-Text VQA dataset, containing VQAv2 [4], TextVQA [41] and ST-VQA [8] for 10 epochs using a batch size of 8 and 16 for ALBEF and BLIP respectively. Moreover, we set $\alpha_1 = \alpha_2 = 1$ and keep the other training-related hyperparameters as in the original papers.

Image Captioning We train UniTNT_{BLIP} on a unified Text-Non-Text CAP dataset, comprised of COCO Captions [9] and TextCaps [40], for 5 epochs with batch size of 32. We set $\alpha_1 = 0.05$ and $\alpha_2 = 0$ since contrary to VQA, CAP does not contain textual information available both in training and inference time, making it infeasible to implement OCR-BC. Moreover, we keep the rest of the hyperparameters as in BLIP.

A.3. Scene-text information

As specified in the paper, we extract the scene-text information (word tokens and 2-dimensional position) for all the VQA and CAP datasets (both the general and scene-text counterparts) using Amazon Text-in-Image. To better understand the prevalence of OCR in the non-scene-text datasets, we plot the statistics of OCR in VQAv2 in Fig. 5 (same images are in COCO Captions as well). While a large portion of the images does not contain text in them, there is a large amount of such with OCR (38.36% and 38.03% of...
train and test images contain OCR). Since OCR conveys meaningful information, it sheds light on the significant improvement of UniTNT up his baselines (ALBEF and BLIP).

B. Datasets

B.1. Visual Question Answering

VQAv2 contains 204,721 images (82,783, 40,504, and 81,434) from COCO [34], 1,105,904 questions (443,757, 214,354, and 447,793), and 6,581,110 answers (4,437,570, 2,143,540, and the test answers are held-out). Answering the questions requires vision-language understanding and commonsense knowledge. Each question has ten ground-truth answers.

TextVQA contains 28,408 images from OpenImages [28], 45,336 questions and 453,360 ground-truth answers. The annotators were instructed to formulate questions that require reasoning from the text in the image. As in VQAv2, each question has 10 ground-truth answers.

ST-VQA is a fusion of computer-vision datasets — ImageNet [14], VizWiz [6], Visual Genome [27], IIIT Scene Text Retrieval [37], ICDAR 2013 [25], ICDAR 2015 [24] and COCO Text [44]. It contains 31K questions, split into training (26K) and testing (5K), requiring scene-text understanding.

B.2. Image Captioning

COCO Captions contains over one and a half million captions describing over 330,000 images from the COCO dataset. Each image has five human-generated captions.

TextCaps is composed of 28,408 images and 142,040 captions (5 captions per image). The images are from the TextVQA dataset, and the captions are based on the text in the image. Specifically, models have to reason over the scene-text information to generate correct captions.

C. The Impact of Training Data

In this section, we study the effect of the different combinations of training datasets and report our findings in Tab. 7. In particular, we experiment with UniTNT and BLIP in Visual Question Answering and Image Captioning using separate training on vision-oriented and OCR-oriented datasets and combined training. In VQA, using both dataset types leads to the best standalone and average performance in the tested benchmarks. This attests to the symbiosis between general and scene-text-oriented VQA, encouraging avoidance of the common practice of separate finetuning.

However, using a unified training set in CAP leads to the best COCO Captions and average results, but not in TextCaps. Specifically, separate finetuning on TextCaps achieves a CIDEr score of 130.5, compared to 119.1 in the combined training. It corresponds with [40], which shows that combining COCO Captions with an upsampled version of TextCaps reduces the model’s performance on the former. It is because while training on TextCaps encourages the model to insert OCR into the caption, training on COCO Captions which barely contains OCR in its captions, penalizes such behavior, leading to an intrinsic tradeoff. To better understand the effects of training models solely on TextCaps, we qualitatively test them on COCO Captions. Notably, we finetune both BLIP and UniTNT of TextCaps and demonstrate their performance on COCO Captions in Fig. 6. Our analysis shows that as TextCaps contains OCR in all its captions, separate finetuning causes models to fixate on OCR, regardless of their importance. Moreover, in images without an OCR signal, the models sometimes hallucinate text in the image. While both models showcase similar behavior, since UniTNT has better scene-text understanding, it is more prone to such phenomena. It is also expressed in Tab. 7, where BLIP and UniTNT trained on TextCaps obtain 84.8 and 70.4 on COCO Captions, respectively. Despite the improved performance on TextCaps when performing separate finetuning on it, our findings highlight its drawbacks. Thus, we claim that also in CAP, combined training should be applied.

From a general view, we hypothesize that since numerous valid captions exist for a given image, both with and
without OCR, the model struggles to decide whether to use the OCR in its caption. Due to the datasets’ sizes in combined training that favors the vision-oriented ones, the model opts to reduce its use of OCR, not fully maximizing its performance on TextCaps. It is contrary to VQA, where the conditioning over the question makes it easier for the model to decide whether to use OCR or not (e.g., "What is written in the sign?" versus "What color is this shirt?").

D. Qualitative Analysis

Visual Question Answering  We provide an additional qualitative demonstration of UniTNT and compare it to BLIP and M4C on both TextVQA validation set (Fig. 7) and VQAv2 test set (Fig. 8). We depict in the four leftmost columns success-cases and the rightmost, fail cases, and color in green the correct answers and red, incorrect ones. Moreover, we divide the figures such that the upper part corresponds with the benchmark’s goal (VQAv2 – see, TextVQA – read) and the lower one with the counterpart goal (VQAv2 – read, TextVQA – see). These results further demonstrate that UniTNT is capable of reasoning over both visual and scene-text information, while other competing methods perform unsatisfactorily on at least one of the benchmarks. Moreover, as the visualizations in Fig. 8 testify, granting scene-text understanding also benefit VQAv2, corresponding with the quantitative evidence in the main paper. It is demonstrated in the bottom part of the figure, where the OCR is crucial for answering the questions or providing meaningful information that facilitates answering them.

Image Captioning  Similar to the VQA demonstration, we present a visualization of UniTNT performance on TextCaps (Fig. 9 and COCO Captions (Fig. 10) and compare the performance to M4C and BLIP. On the left columns, we show images where our method outperforms the other methods, and on the right, its failure cases. Moreover, we list the CIDEr scores of each prediction and color in green the highest one. These findings attest that BLIP is incapable of incorporating scene-text information, which results in relatively low CIDEr results. Interestingly, M4C is too overfitted for TextCaps, causing it to fail completely on COCO Captions where OCR is scarce. Specifically, it focuses on the OCR regardless of their importance (e.g., the third example in the last row of Fig. 10) and thus provides an irrelevant caption. Despite the intrinsic tradeoff described in the paper between TextCaps and COCO Captions, UniTNT is capable of providing adequate captions for both benchmarks. Specifically, our method is the only one to cope satisfactorily on both benchmarks altogether and is capable of harnessing both scene-text and visual information.
Figure 6: **Qualitative demonstration of the effects of finetuning on TextCaps.** BLIP and UniTNT results of COCO Captions when finetuned solely on TextCaps. In some cases, scene-text understanding helps the models, but it also leads to over-reliance on the OCR signal and even to the hallucination of OCR. While such phenomena occur in both models, it is more prevalent in UniTNT due to its better scene-text understanding.
Figure 7: Qualitative demonstration on TextVQA validation. UniTNT, M4C, and BLIP answers, containing both success (left) and fail (right) cases of our method on image-question pairs that require mainly reading (top) and ones that require also visual reasoning (bottom).
Figure 8: Qualitative demonstration on VQAv2 test. UniTNT, M4C, and BLIP answers, containing both success (left) and fail (right) cases of our method on image-question pairs that require mainly vision (top) and ones that require also scene-text understanding (bottom).
Figure 9: **Qualitative demonstration on TextCaps.** UniTNT, M4C-Captioner, and BLIP answers, containing both success (left) and fail (right) cases of our method alongside the per-caption CIDEr score.
Figure 10: Qualitative demonstration on COCO Captions. UniTNT, M4C-Captioner, and BLIP answers, containing both success (left) and fail (right) cases of our method alongside the per-caption CIDEr score.