

Towards Unified Scene Text Spotting based on Sequence Generation

Taeho Kil^{1*} Seonghyeon Kim^{2†} Sukmin Seo¹ Yoonsik Kim¹ Daehee Kim¹

¹Naver Cloud ²Kakao Brain

{taeho.kil, sukmin.seo, yoonsik.kim90, daehee.k}@navercorp.com, matt.mldev@kakaobrain.com

Abstract

Sequence generation models have recently made significant progress in unifying various vision tasks. Although some auto-regressive models have demonstrated promising results in end-to-end text spotting, they use specific detection formats while ignoring various text shapes and are limited in the maximum number of text instances that can be detected. To overcome these limitations, we propose a UNified scene Text Spotter, called UNITS. Our model unifies various detection formats, including quadrilaterals and polygons, allowing it to detect text in arbitrary shapes. Additionally, we apply starting-point prompting to enable the model to extract texts from an arbitrary starting point, thereby extracting more texts beyond the number of instances it was trained on. Experimental results demonstrate that our method achieves competitive performance compared to state-of-the-art methods. Further analysis shows that UNITS can extract a larger number of texts than it was trained on. We provide the code for our method at <https://github.com/clovaai/units>.

1. Introduction

End-to-end scene text spotting, which can jointly detect and recognize text from an image at once, has recently gained significant attention. This work has practical applications in various fields such as visual navigation, visual question answering, and document image understanding.

In this paper, we formulate scene text spotting as a sequence generation task. This approach casts the vision task as a language modeling task conditioned on the image and text prompt [6, 7, 27, 38]. By formulating the output as a sequence of discrete tokens, the vision task can be performed by generating a sequence through an auto-regressive transformer decoder. Recent studies have attempted to integrate various tasks into an auto-regressive model, including text spotting. SPTS [28] treated all detection formats as the sin-

*Corresponding author.

†This work was done while the authors were at Naver Cloud.

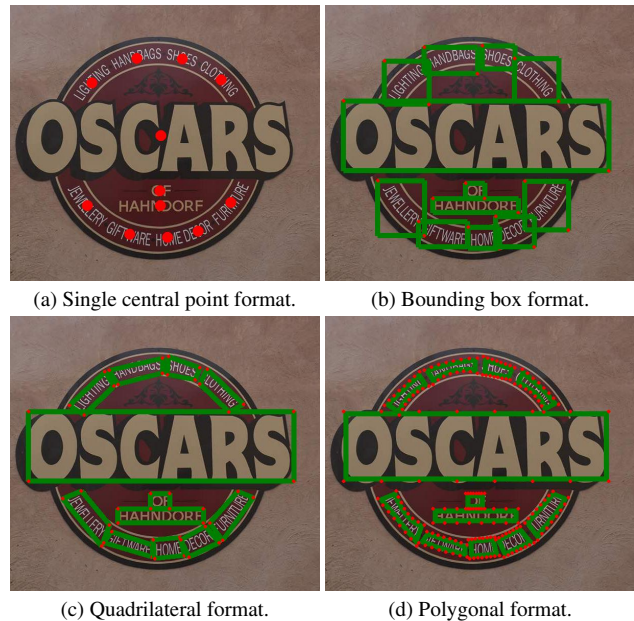


Figure 1. Various types of detection formats. The green line represents the boundary shape of the detection format, and the red dot represents the points used for the corresponding format.

gle central point and predicted the coordinate tokens of the central point and word tokens auto-regressively. However, there are several challenges associated with applying the sequence generation approach directly to scene text spotting.

As illustrated in Fig. 1, there are various methods to indicate the location and boundaries of text instances, and each method has its own trade-offs in terms of annotation costs and potential applicability. For instance, a single central point or bounding box annotation has a relatively low annotation cost and is appropriate when the detailed shape information is not necessary. However, in many fields where text spotting is applied, handling text location information as only a single point might be insufficient. As shown in Fig. 2, in the scene text editing [17, 32], which converts the text in the image to desired text while preserving the text style, detailed text shape extraction is required. Therefore, for such

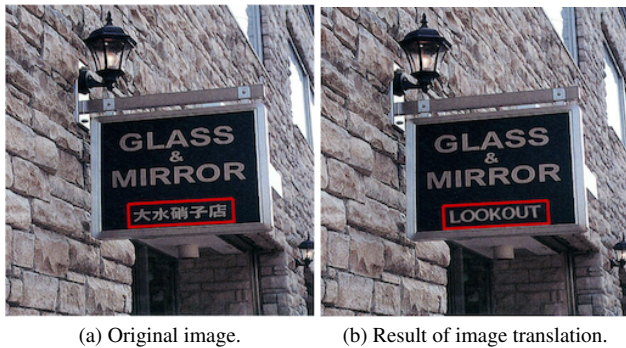


Figure 2. Example of image translation using scene text editing.

tasks, outputting text location information in quadrilateral or polygonal formats beyond a single point is necessary. Similarly, in visual document understanding, which utilizes optical character recognition as a pre-processing step, more detailed detection information of texts could be useful. In summary, since each detection format has its own trade-offs, it is better to cover all detection formats instead of relying on only one.

Also, conventional sequence generation models have a limitation in terms of generating sequences longer than the maximum length the model has been trained on. This limitation is particularly problematic for scene text spotting, as it requires generating more points than bounding boxes and generating text transcriptions in addition to object classes. For text-rich images such as documents, this limitation becomes a bottleneck in extracting all the texts.

To address these challenges in applying the sequence generation method to text spotting, we propose a novel end-to-end UNified scene Text Spotter, called UNITS, which aims to overcome these limitations. We unify various detection formats, such as quadrilateral and polygon, into the model, allowing it to detect arbitrary-shaped text areas. This approach enables the model to learn different annotation formats together using a single unified model, and it can extract all detection formats. We use prompts to unify various detection formats, enabling a single model to predict the coarse or fine-grained location information of text instances.

Moreover, we introduce starting-point prompting, which enables the model to extract texts from arbitrary starting points, allowing it to generate longer sequences than the maximum decoding length. This approach enhances the model’s efficiency and enables it to extract more text than the number of text boxes it has been trained on.

In addition, we employ a multi-way transformer decoder from the mixture-of-experts (MoE) [4, 9, 37]. This decoder separates each time stamp into detection and recognition, allowing the model to converge faster by guiding the model to generate a detection or a recognition token and assigning

an expert accordingly. Each block of the multi-way transformer consists of shared attention modules and two task experts: detection and recognition experts.

Experimental results demonstrate that our proposed method achieves competitive performance on text spotting benchmarks while extracting texts in various detection formats using a single unified model.

Our main contributions are:

- We propose a novel sequence generation-based scene text spotting method that can extract arbitrary-shaped text areas by unifying various detection formats.
- Our model can extract more texts than the decoder length allows using the starting-point prompt, which generalizes the model to spot more texts than it has been trained on.
- Experimental results demonstrate that our method achieves competitive performance on text spotting benchmarks and provides additional functionalities

2. Related Work

Text Spotting. The text spotting that detects and recognizes text at once has recently attracted a lot of attention. Many studies improved the performance by learning the detector and recognizer simultaneously [3, 10, 12, 15, 16, 18, 22, 23, 28, 29, 36, 39], rather than separately [5, 19, 20, 33]. Rosetta++ [5] detected text in the form of a bounding box. Some methods [19, 22, 40] treated text location information with the quadrilateral format and detected inclined text beyond horizontal text. For arbitrary-shaped text detection, some works used parametric Bezier curve representations [23, 24, 39]. The segmentation-based method [18, 21, 29, 29] can handle all text instances of various shapes, however, there is a disadvantage that it needs complex post-processing such as connected components filtering. Some methods [2, 3, 10, 18, 29, 36] used character-level as well as word-level annotations. It enables the detection of arbitrarily-shaped texts, however, character-level annotation is expensive and it requires post-processing like word grouping.

Recently, there were also attempts to develop a model that can deal with various detection formats. DEER [15] recognized words by guiding text instance location as a simple single reference point without RoI cropping. TTS [16] proposed a weakly supervised learning method that makes the box, and mask predictors lightweight and treated various types of annotation formats while sharing the backbone and encoder. SwinTextSpotter [12] and MaskTextSpotter [18] also used separated task-specific heads corresponding to different annotation formats. Although the backbone and encoder are shared in these works, the task-specific heads are separated by detection formats so the proposed method is a more strictly unified approach. The proposed method

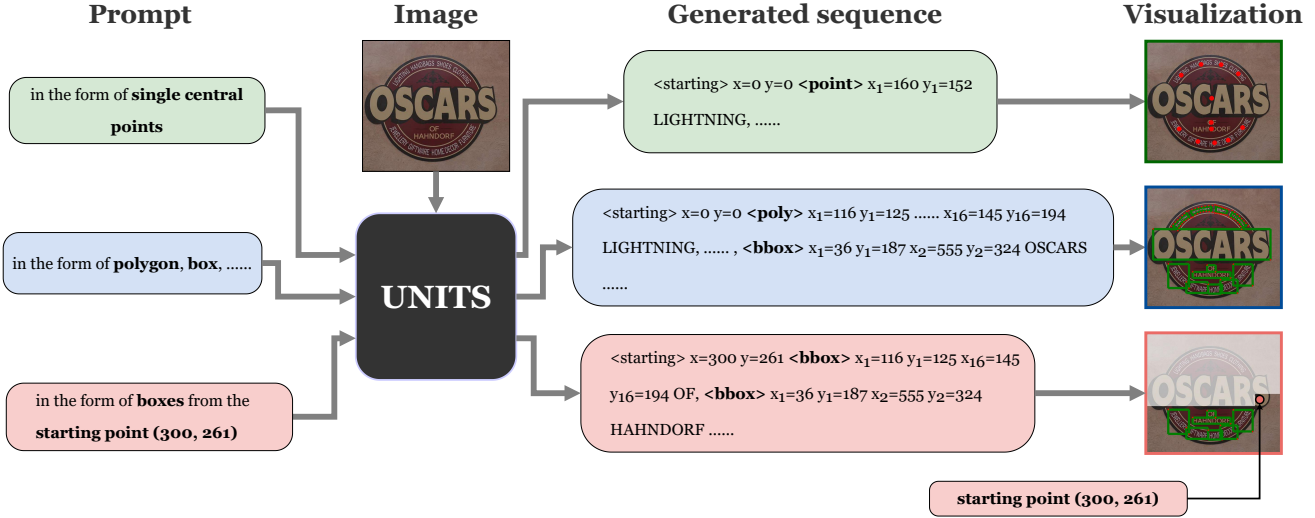


Figure 3. Pipeline of the proposed text spotting method. The information of each instance in the sequence consists of the detection format token, coordinate tokens for detection, and text transcriptions. Coordinate tokens are generated according to the desired detection format. The model sequentially detects and recognizes only text instances located after a specific location determined by the starting-point prompt.

can learn different annotation types at once and extract the location information of the texts into the desired detection formats.

Sequence Generation-based Approach. Recently, for vision tasks, the methods to apply the seq2seq approach used in NLP have been studied. Instead of designing a model specific to the vision task, this approach uses a simple transformer encoder and decoder structure and performs the task by predicting sequence. Pix2seq [6] was the first attempt at this approach, it showed that objects can be detected well even in random order without any matching algorithm like Hungarian matching. UniTAB [38] performed multi-task learning of vision language tasks such as object detection, captioning, and visual question answering by utilizing the method of guiding tasks through task prompts like T5 [30]. Unified-IO [27] further jointly trained various vision language tasks including image generation by using the unified input/output formats for all tasks. The generation-based methods can unify various tasks and modalities in simple transformer encoder-decoder architecture.

There was a study in which a sequence generation-based method has been applied for text spotting. SPTS [28] treated all location information as the central point and predicted the coordinate tokens of the central point and word transcription tokens auto-regressively. This method showed that single point location and transcription for text instances are well generated, however, it can not handle the more complex detection formats. Also, it has the limitations about decoder length that common sequence generation methods have. This method cannot handle the case where the number of objects is larger than the number of objects allowed by the decoder length. The proposed method

can handle the various types of detection formats through prompts and can extract texts beyond the number of text instances that have been trained.

3. Method

We propose an end-to-end scene text spotting method, named UNITS, which is based on a unified interface that handles various text detection formats and determines the starting point of text to extract. The overall pipeline is shown in Fig. 3. The details of the unified interface are described in 3.1, while the architecture of the proposed method, including the multi-way transformer decoder and objective function, is presented in 3.2.

3.1. Unified Interface for Text Spotting

The sequence construction method of UNITS follows recent methods [6,7,27,28,38]. The location and transcription of text instances are converted into discrete tokens through tokenization and concatenated in each instance. As the location token, the x, y coordinates of the point are normalized to the width and height of the image, and then quantized to $[0, n_{bins}]$. The bin size n_{bins} means the number of coordinate vocab, we set the bin size as 1,000.

Given the significant differences in detection formats, existing methods use task-specific heads. To handle various format types with only a single model, we introduce a new sequence interface by adding prompts for each text instance. Before generating the output of each text instance, the model determines the shape of the text area to be extracted through a prompt. This prompt named detection format token is fed as input, and the model can extract

the text instance in the desired format corresponding to the prompt. As shown in Fig. 3, when predicting texts in the form of the polygon, we insert the detection format token `<poly>` before location tokens, and when treating bounding box-shaped instances, we insert the detection format token `<bbox>`. After the detection format token, location information is extracted according to this format, and text transcription is generated next.

If there are more text instances in the image than the number of instances determined by the decoder length, it can cause a problem where the sequence generation model cannot detect and recognize all texts in the image. To overcome this problem, we introduce starting-point prompting, which can guide the image region where text should be detected and can extract texts beyond the number of text instances that the model has been trained on.

Detection Format Token. The use of detection format tokens enables a single model to handle multiple detection formats using a shared vocabulary. The special tokens and the resulting output sequence are illustrated in Fig. 4. There are five texts, and they are arranged in a 1D sequence using raster scan order as ‘OLD’, ‘MILL’, ‘RECEPTIO’, ‘HOTEL’, and ‘MOTEL’. For each text instance, the detection format token determines whether it should be represented as a bounding box or another format, such as a polygon or a single point. In the second turn, the detection format token corresponding to the box annotation is used, resulting in the extraction of the ‘MILL’ instance as a bounding box.

In this paper, we use a total of four detection format tokens: central point, bounding box, quadrilateral, and polygonal formats. The central point, bounding box, quadrilateral, and polygonal formats are expressed as 1, 2, 4, and 16 points, respectively, and the length of transcriptions is fixed at 25. When constructing transcription tokens, if the text length exceeds 25, the excess part is removed, while if the text length is shorter, it is padded. Through this sequence construction configuration, we propose a unified interface that can handle multiple detection formats simultaneously.

Starting-Point Prompting. Multiple text instances are generated in raster scan order, and sequence generation is guided by the starting-point prompt. Thus, UNITS can read texts from a specific location in raster scan order by giving a specific starting point. This makes it possible to detect a large number of texts, such as texts in a document, even with a limited decoder length.

As shown in Fig. 5, only texts located after the starting point are detected and recognized in raster scan order. This strategy teaches the model to detect text instances from an arbitrary starting point while ignoring the text instances before the starting point. The starting-point prompt composes of the x, y coordinates of the point. When testing, first we set the starting point as the top-left point of the image, then a sequence of text instances in front is first generated. If

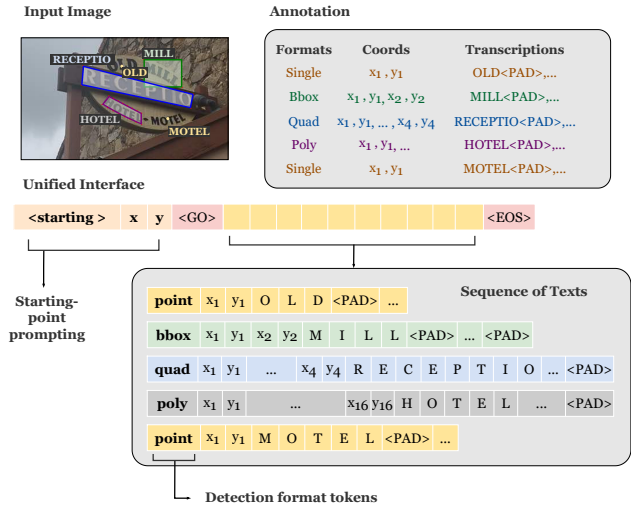


Figure 4. Illustration of unified interface for text spotting.

the generated output sequence does not end with an `<eos>` token, our method sets the starting point as the last detected text position in the previous step, then continues and enables the re-generation of a text sequence corresponding to the remaining objects. By repeating this process until `<eos>` token appears, the model is able to cover whole images in raster scan order, and all texts in the image can be detected and recognized by concatenating parts of texts. In the training step, the starting point in the image is set randomly so that the model learns the ability to detect text instances from the arbitrary starting point in the image.

3.2. Architecture and Objective

Architecture. Our model follows a basic encoder-decoder structure, where the text spotting task is cast as a language modeling task conditioned on the image, similar to conventional approaches [6, 7, 27, 28, 38]. For encoding the raw image, we use the Swin Transformer [25] as the image encoder. To generate the text sequence, we use a Transformer decoder that generates one token at a time conditioned on the previous tokens and encoded image features.

Multi-way Decoder. We use a multi-way transformer decoder to handle the learning of multiple detection formats simultaneously. Since each detection format has a different number of location tokens, learning multiple formats with varying patterns in the sequence is a more challenging task than learning a single format with a fixed pattern. The multi-way decoder uses strong supervision by indicating whether it is time to generate detection or recognition tokens, thus helping to alleviate the difficulty of our model.

Each block of the multi-way transformer decoder consists of shared attention modules and two task experts: the detection expert and the recognition expert. When generating the sequence, our model generates detection informa-

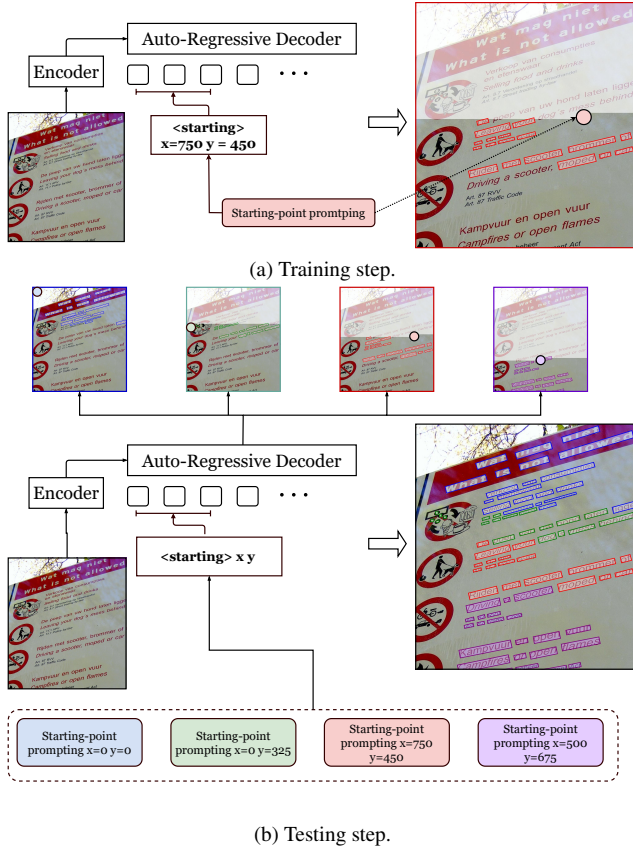


Figure 5. Illustration of starting-point prompting. In the training step, the model learns the ability to extract texts after a randomly selected starting point. In the testing step, the model extracts all texts by using multiple prompts.

tion for text instances first, followed by transcription information. This allows the two tasks to be separated by the time stamp of the sequence. Each feedforward neural network (FFN) expert helps the single unified model to focus on detection and recognition, respectively.

Objective. Since UNITS predicts tokens through a decoder without a task-specific head, we train the model with a standard cross-entropy loss. At each time stamp j , the model aims to maximize the likelihood of the target tokens \tilde{y} given the input image I and previously generated tokens $y_{1:j-1}$:

$$L = - \sum_{j=1}^N w_j \log P(\tilde{y}_j | I, y_{1:j-1}), \quad (1)$$

where y and \tilde{y} denote the input and target sequence, respectively, and N is the sequence length of sequence. Also, w_j is the weight value for the j -th token, 0 is assigned at prompt token to exclude it from the loss calculation, and 1 is assigned for the remaining tokens.

4. Experiments

4.1. Dataset

We use the following datasets for the experiments: **Curved SynthText** [24] is a synthetic dataset consisting of 94,723 images including mostly straight texts and 54,327 images including curved texts. It is annotated in bezier annotation formats. **ICDAR 2013** [14] is a scene text dataset containing most of the well-captured and horizontal texts, annotated in bounding box formats. It contains 229 train and 233 test images. **ICDAR 2015** [13] is a scene text dataset containing inclined texts, including 1,000 train and 500 test images. It is annotated in quadrilateral formats. **MLT 2019** [1] is a dataset containing 10,000 train and 10,000 test images. It contains texts from a total of 7 languages and is annotated in quadrilateral formats. **Total-Text** [8] is a dataset including arbitrary-shaped texts. All are annotated in quadrilateral or polygonal formats and it contains 1,255 train and 300 test images. **TextOCR** [34] is a relatively recently published dataset to benchmark text recognition. All images are annotated in quadrilateral or polygonal formats, and it is composed of 21,749 train, 3,153 validation, and 3,232 test images. **HierText** [26] is a most recently published dataset featuring hierarchical annotations of text in natural scenes and documents. It is annotated in polygonal or quadrilateral annotation formats. It contains 8,281 train, 1,724 validation, and 1,634 test sets.

4.2. Implementation Details

We use Swin-B [25] as the visual encoder of UNITS. The layer numbers and window size are set to $\{2, 2, 18, 2\}$ and 7, respectively. We use the initial model pre-trained by ImageNet 22k dataset. For the decoder of UNITS, we use 8 transformer decoder layers. The maximum length in the decoder is set as 1024, and the decoder weights are randomly initialized.

The model is first pre-trained on a combination of datasets including Curved SynthText, ICDAR 2013, ICDAR 2015, MLT 2019, Total-Text, TextOCR, and HierText. To ensure the model can learn all detection formats well, 40% of the annotations are randomly converted into bounding box or central point formats. We use a batch size of 128, and the model is pre-trained for 300k steps, with initial 10k warm-up steps. We use AdamW optimizer with a cosine learning rate scheduler, and set the learning rate as $3e^{-4}$ and the weight decay is set to $1e^{-4}$. For data augmentation, we apply random rotation, random resize between 30% and 200%, random crop, and color jitter. The image is resized to 768×768 after being padded in a square shape. After pre-training, the model is fine-tuned on the same combination of datasets used in pre-training for 120k steps. We use a batch size of 16 and set the input image resolution as 1920×1920 . In fine-tuning, the AdamW optimizer with cosine learning

Method	Detection			End-to-End			
	Recall	Precision	F-measure	Strong	Weak	Generic	None
CRAFTS [3]	85.3	89.0	87.1	83.1	82.1	74.9	-
MaskTextSpotter v3 [18]	-	-	-	83.3	78.1	74.2	-
ABCNet v2 [23]	86.0	90.4	88.1	82.7	78.5	73.0	-
MANGO [29]	-	-	-	85.4	80.1	73.9	-
DEER [15]	86.2	<u>93.7</u>	89.8	82.7	79.1	75.6	71.7
SwinTextSpotter [12]	-	-	-	83.9	77.3	70.5	-
TESTR [39]	89.7	90.3	90.0	85.2	79.4	73.6	65.3
TTS [16]	-	-	-	85.2	81.7	77.4	-
GLASS [31]	-	-	-	84.7	80.1	76.3	-
UNITS _{Shared}	<u>90.5</u>	93.6	<u>92.0</u>	<u>88.4</u>	<u>83.9</u>	<u>79.7</u>	<u>78.5</u>
UNITS	91.0	94.0	92.5	89.0	84.1	80.3	78.7

Table 1. Experiment results on ICDAR 2015. “Strong”, “Weak”, “Generic”, and “None” represent recognition with each lexicon respectively.

Method	Detection	End-to-End	
	F-measure	None	Full
CRAFTS [3]	87.4	78.7	-
MaskTextSpotter v3 [18]	-	71.2	78.4
ABCNet v2 [23]	87.0	70.4	78.1
MANGO [29]	-	72.9	83.6
DEER [15]	85.7	74.8	83.3
SwinTextSpotter [12]	88.0	74.3	84.1
TESTR [39]	86.9	73.3	83.9
TTS [16]	-	75.6	84.4
GLASS [31]	-	76.6	83.0
UNITS _{Shared}	<u>88.4</u>	77.3	<u>85.0</u>
UNITS	89.8	78.7	86.0

Table 2. Experiment results on Total-Text. “Full” and “None” represent recognition with each lexicon respectively.

Method	End-to-End		
	Strong	Weak	Generic
SPTS [28]	77.5	70.2	65.8
UNITS _{Shared} – Point	89.9	84.1	79.3
UNITS _{Shared} – Box	90.1	84.5	79.3
UNITS _{Shared} – Quad	89.9	84.5	79.5
UNITS _{Shared} – Polygon	89.4	84.0	79.0

Table 3. The end-to-end recognition performance evaluated by the point-based metric [28] on ICDAR 2015.

rate scheduler is used with the learning rate $1.2e^{-4}$. Additionally, since minor annotation rules are different for each dataset, we fine-tune one more model for each benchmark

dataset. Each model is fine-tuned for 20k steps with a fixed learning rate $3e^{-5}$, and we report the results of unified and dataset-specific models.

4.3. Results on Text Spotting Benchmarks

We evaluate the performance of UNITS and existing methods on ICDAR 2015 and Total-Text datasets. For ICDAR 2015, we directly predict quadrilaterals and for Total-Text, we use 16-point polygons. Both detection and end-to-end results are reported in Tab. 1 and Tab. 2. For ICDAR 2015, we use “strong”, “weak”, and “generic” dictionaries, and also evaluate “None” that doesn’t use any dictionary. For Total-Text, we show the results that are “Full” with a lexicon and “None” without any lexicon. UNITS_{Shared} is the unified model, and the other is the model which is fine-tuned on each dataset.

On both benchmark datasets, the proposed method shows state-of-the-art performance for all vocabularies in the end-to-end evaluation protocol. We confirm that our method is a competitive method compared to the existing methods for both quadrilateral and polygonal output formats. Some qualitative results are shown in Figs. 6 and 7, where the unified model UNITS_{Shared} extracted texts in the desired detection formats.

In addition, we compare the proposed method with the previous study SPTS [28], which is also a sequence generation method that only handles the single central point format. SPTS proposed a point-based evaluation metric for evaluating single central point output, and we use the same metric for performance evaluation. The evaluation results are shown in Tab. 3. Besides single point output, UNITS also can output in the box, quadrilateral, and polygonal formats. To show the end-to-end spotting performance of all output formats, we calculate the central point for all output



Figure 6. Qualitative results of our method on ICDAR 2015 and Total-Text.



Figure 7. Prediction examples of our unified model. The proposed model can extract text in several detection formats with a single model.

formats and measure the performance with the point-based metric. It shows better performance than the existing SPTS for all detection formats, including central points. The experimental results for other benchmark datasets are shown in the supplementary material.

4.4. Ablation studies

We conduct ablation experiments on ICDAR 2015, Total-Text, and TextOCR to evaluate the effectiveness of the proposed method. These experiments do not use a lexicon and follow the end-to-end evaluation protocol.

Image Encoder. We compare the swin transformer with the hybrid transformer, which is a combination of ResNet-50 [11] and vanilla transformer [35], as image encoders. For the swin transformer, we resize the longer side of the input image to 1920 during both training and inference. However, the computational complexity of self-attention is quadratic to the image size, which leads to memory issues when training a hybrid transformer at high resolution. Therefore, we set the longer side of the input image as 1280 when using the hybrid transformer. The experimental results are shown

Method	ICDAR 2015		Total-Text	
	DET	E2E	DET	E2E
Hybrid transformer	89.6	65.6	85.9	67.4
Swin transformer	92.0	78.5	88.4	77.3

Table 4. Ablation study of the image encoder on ICDAR 2015 and Total-Text. A window attention-based swin transformer capable of high-resolution training and testing shows better performance than the hybrid transformer.

Starting-Point Prompt	End-to-End		
	Precision	Recall	F-measure
-	78.4	30.6	44.0
✓	80.2	54.2	64.7

Table 5. Ablation study of the starting-point prompting on TextOCR. The starting-point prompting enables UNITS to extract a large number of text instances even with a limited decoder length.

Decoder Length	End-to-End		
	Precision	Recall	F-measure
1024	80.2	54.2	64.7
512	78.2	53.7	63.7
256	76.8	51.8	61.8

Table 6. Performance of UNITS on TextOCR with different decoder lengths. The starting-point prompting works robustly regardless of decoder length, allowing UNITS to detect more text instances than the decoder length allows.

Method	ICDAR 2015		Total-Text	
	DET	E2E	DET	E2E
Vanilla decoder	73.7	39.1	76.1	45.1
Multi-way decoder	75.1	40.7	78.9	47.7

Table 7. Ablation study of the multi-way decoder on ICDAR 2015 and Total-Text. Both detection and end-to-end performance increase when a multi-way decoder is applied.

in the Tab. 4. In particular, the hybrid transformer shows lower performance. We confirm that a high-resolution input image is essential for small text detection and recognition, and for this, the swin transformer, which has linear computational complexity to the image size, is a better choice than a vanilla transformer.

The Starting-Point Prompting. The proposed method can detect and recognize more text instances than the maximum number of instances allowed by the decoder length through the starting-point prompting. To demonstrate this effect, we perform an ablation study by comparing the performance of UNITS with and without the starting-point prompt. We conduct the experiment on the TextOCR dataset, which contains a relatively large number of texts, and evaluate the model using the point-based metric proposed in [28]. In the experiment, we predict a single central point, so the method without using the prompt is similar to the existing method [28]. The results are shown in Tab. 5, where we observe that starting-point prompting significantly improves the recall of the model, as it can detect more instances beyond the maximum sequence lengths. To test the robustness of starting-point prompting, we also evaluate UNITS under three different decoder lengths and compare the results. We present the performance in Tab. 6 and confirm that the starting-point prompting method works robustly regardless of the decoder length.

As shown in Fig. 8, starting-point prompting continuously detects and recognizes objects even with a limited decoder length, by taking the starting point as the location of the last detected text instance. In contrast, existing sequence generation-based methods [28] have a limitation in extract-



(a) The existing method [28]. (b) Proposed method.

Figure 8. Visualization results of the existing sequence generation-based method [28] and our method. Our method overcomes the limitations of existing methods by using the starting-point prompt.

ing a large number of texts. Our method alleviates this limitation and allows the extraction of texts beyond the number of text boxes that have been trained on.

Multi-way Transformer Decoder. We adopt a multi-way transformer decoder structure in which detection and recognition FFN experts are separated to improve the detection and recognition performance of UNITS. To verify the effectiveness of the multi-way transformer decoder structure, we compare it with the vanilla transformer decoder. In this ablation study, we use a hybrid transformer as a baseline encoder and train both models from scratch with a longer side of the input image of 768, and test them with an image size of 1280. The results in Tab. 7 demonstrate that both the detection and end-to-end performance are improved when the multi-way decoder is applied. We hypothesize that the multi-way decoder converges faster by determining in advance whether it is the turn to generate a detection token or a recognition token and assigning an expert FFN accordingly.

5. Conclusions

In this paper, we have presented a novel end-to-end scene text spotting method that unifies various detection formats to extract arbitrary-shaped text areas. To handle multiple detection formats simultaneously, we employ a multi-way transformer decoder, and the starting-point prompting strategy enables our model to extract a larger number of texts than seen in training data. Experimental results show that our method achieves competitive results on benchmark datasets, outperforming text-spotting-specific models even with a simple sequence generation model. The success of our method suggests its potential applicability in other vision tasks, such as object detection or segmentation.

References

- [1] Saadullah Amin, Günter Neumann, Katherine Dunfield, Anna Vechkaeva, Kathryn Annette Chapman, and Morgan Kelly Wixted. Mlt-dfki at clef ehealth 2019: Multi-label classification of icd-10 codes with bert. In *CLEF (Working Notes)*, pages 1–15, 2019. [5](#)
- [2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoon Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019. [2](#)
- [3] Youngmin Baek, Seung Shin, Jeonghun Baek, Sungrae Park, Junyeop Lee, Daehyun Nam, and Hwalsuk Lee. Character region attention for text spotting. In *European Conference on Computer Vision*, pages 504–521. Springer, 2020. [2](#), [6](#)
- [4] Hangbo Bao, Wenhui Wang, Li Dong, and Furu Wei. V1-beit: Generative vision-language pretraining. *arXiv preprint arXiv:2206.01127*, 2022. [2](#)
- [5] Fedor Borisjuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 71–79, 2018. [2](#)
- [6] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021. [1](#), [3](#), [4](#)
- [7] Ting Chen, Saurabh Saxena, Lala Li, Tsung-Yi Lin, David J Fleet, and Geoffrey Hinton. A unified sequence interface for vision tasks. *arXiv preprint arXiv:2206.07669*, 2022. [1](#), [3](#), [4](#)
- [8] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017. [5](#)
- [9] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2021. [2](#)
- [10] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9076–9085, 2019. [2](#)
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [7](#)
- [12] Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4593–4603, 2022. [2](#), [6](#)
- [13] Dimosthenis Karatzas, Lluís Gomez-Bigorda, Angelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. [5](#)
- [14] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluís Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluís Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th international conference on document analysis and recognition*, pages 1484–1493. IEEE, 2013. [5](#)
- [15] Seonghyeon Kim, Seung Shin, Yoonsik Kim, Han-Cheol Cho, Taeho Kil, Jaehung Surh, Seunghyun Park, Bado Lee, and Youngmin Baek. Deer: Detection-agnostic end-to-end recognizer for scene text spotting. *arXiv preprint arXiv:2203.05122*, 2022. [2](#), [6](#)
- [16] Yair Kittenplon, Inbal Lavi, Sharon Fogel, Yarin Bar, R Manmatha, and Pietro Perona. Towards weakly-supervised text spotting using a multi-task transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4613, 2022. [2](#), [6](#)
- [17] Junyeop Lee, Yoonsik Kim, Seonghyeon Kim, Moonbin Yim, Seung Shin, Gayoung Lee, and Sungrae Park. Rewrittenet: Realistic scene text image generation via editing text in real-world image. *arXiv preprint arXiv:2107.11041*, 2021. [1](#)
- [18] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *European Conference on Computer Vision*, pages 706–722. Springer, 2020. [2](#), [6](#)
- [19] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018. [2](#)
- [20] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-first AAAI conference on artificial intelligence*, 2017. [2](#)
- [21] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11474–11481, 2020. [2](#)
- [22] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018. [2](#)
- [23] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9809–9818, 2020. [2](#), [6](#)
- [24] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *arXiv preprint arXiv:2105.03620*, 2021. [2](#), [5](#)
- [25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

- Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4, 5
- [26] Shangbang Long, Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis. Towards end-to-end unified scene text detection and layout analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1049–1059, 2022. 5
- [27] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Motlaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 1, 3, 4
- [28] Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Jing Li, Shenggao Zhu, Dahua Lin, Chunhua Shen, et al. Spts: Single-point text spotting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4272–4281, 2022. 1, 2, 3, 4, 6, 8
- [29] Liang Qiao, Ying Chen, Zhazhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Mango: A mask attention guided one-stage scene text spotter. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2467–2476, 2021. 2, 6
- [30] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67, 2020. 3
- [31] Roi Ronen, Shahar Tsiper, Oron Anshel, Inbal Lavi, Amir Markovitz, and R Manmatha. Glass: Global to local attention for scene-text spotting. In *European Conference on Computer Vision*, pages 249–266. Springer, 2022. 6
- [32] Prasun Roy, Saumik Bhattacharya, Subhankar Ghosh, and Ummapada Pal. Stefann: scene text editor using font adaptive neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13228–13237, 2020. 1
- [33] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 2
- [34] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Wojciech Galuba, and Tal Hassner. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8802–8812, 2021. 5
- [35] 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. 7
- [36] Pengfei Wang, Chengquan Zhang, Fei Qi, Shanshan Liu, Xiaoqiang Zhang, Pengyuan Lyu, Junyu Han, Jingtuo Liu, Er-rui Ding, and Guangming Shi. Pgnnet: Real-time arbitrarily-shaped text spotting with point gathering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2782–2790, 2021. 2
- [37] Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Vlmto: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*, 2021. 2
- [38] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 521–539. Springer, 2022. 1, 3, 4
- [39] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9519–9528, 2022. 2, 6
- [40] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017. 2