Copyright ©2026 IEEE. Reprinted from B. Clipp, N. Damer, S. X. Huang and V. Morariu, editors, Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), March 2026.

TiCLS: Tightly Coupled Language Text Spotter

Leeje Jang[1] Yijun Lin[1] Yao-Yi Chiang[1] Jerod Weinman[2]
[1]University of Minnesota, United States
[2]Grinnell College, United States
{jang0124, lin00786, yaoyi}@umn.edu jerod@acm.org

Abstract

Scene text spotting aims to detect and recognize text in real-world images, where instances are often short, fragmented, or visually ambiguous. Existing methods primarily rely on visual cues and implicitly capture local character dependencies, but they overlook the benefits of external linguistic knowledge. Prior attempts to integrate language models either adapt language modeling objectives without external knowledge or apply pretrained models that are misaligned with the word-level granularity of scene text. We propose TiCLS*, an end-to-end text spotter that explicitly incorporates external linguistic knowledge from a character-level pretrained language model.* TiCLS *contains a pretrained linguistic decoder that fuses visual and linguistic features, enabling robust recognition of ambiguous or fragmented text. Experiments on ICDAR 2015, Total-Text, and CTW1500 demonstrate that* TiCLS *achieves state-of-the-art performance, validating the effectiveness of PLM-guided linguistic integration for scene text spotting. The code is available at* https://github.com/knowledge-computing/TiCLS.

1. Introduction

End-to-end scene text spotting detects and recognizes text from scene images in a unified framework. The visual appearance of scene text varies widely, often featuring unusual fonts, perspective distortion, unconventional typography, and even occlusions [35]. Nevertheless, linguistic patterns remain present in the underlying text. Existing text spotters typically rely on convolution- or transformer-based architectures to extract visual features and use these features to predict text. However, text in scene images often appears degraded (blurred, noisy, or occluded), making it insufficient to rely solely on visual cues.

To address this limitation, some prior work incorporates linguistic context into text spotting. For instance, LM-TextSpotter [39] incorporates a parallel recognition branch explicitly trained for autoregressive language modeling. The language modeling contextualizes each character representation using the representations of surrounding characters within the same word during training. However, this contextualization relies solely on the text available in scene text spotting datasets, since the model lacks the ability to pretrain on large-scale text corpora. As a result, the model fails to exploit abundant unlabeled text data and learns contextual patterns only from the limited training text corpora.

Pretrained language models (PLMs) [5, 13, 20, 30, 31], trained on large-scale corpora such as news articles and books, provide external linguistic knowledge by capturing semantic dependencies across sub-word tokens in well-structured sentences. TextBlockV2 [24] incorporates such external linguistic knowledge by detecting coarse, multi-word text blocks and then applying a PLM-based recognizer to transcribe the content within each text block instead of individual words. This approach initializes the recognizer with an existing PLM, which autoregressively decodes text sequences for each block, thereby producing multi-phrase outputs when a block contains multiple lines or phrases. This method highlights the potential of leveraging PLMs in text spotting, especially for images containing large chunks of text, but challenges remain due to the short and fragmented nature of scene text.

Scene text differs significantly from well-structured, sentence-level language: instances are short, spatially fragmented, and often lack coherent grammatical structure. Simply initializing a recognition decoder with existing PLMs is therefore insufficient, as this initialization does not address the mismatch between sentence-level linguistic modeling and the word-level semantics required for scene text spotting. The key challenge is how to effectively adapt PLMs' linguistic knowledge to scene text spotting so that linguistic cues serve as complementary information to support ambiguous or fragmented visual information.

To bridge this gap, we introduce TiCLS (Tightly Coupled Language Spotter), a novel end-to-end text spotting model that integrates external linguistic knowledge from a pretrained language model (PLM) specifically de-

veloped for scene text. TICLS consists of three main components: a visual feature extractor, a visual decoder, and a linguistic decoder. Following prior transformer-based text spotters [8, 41, 45], the visual feature extractor and decoder learn visual representations of scene text. To provide strong external linguistic knowledge tailored for scene text spotting, we propose a new transformer-based encoder-decoder PLM that we pretrain on a large text corpus with character-level prediction granularity instead of sub-word tokens. We then initialize TICLS's linguistic decoder with this PLM, allowing the model to fuse our external linguistic priors with visual features in a tightly coupled manner to produce more accurate and robust text predictions.

We summarize our contributions as follows:

- We propose TICLS, a novel end-to-end text spotter that explicitly incorporates linguistically rich prior knowledge for scene text spotting.
- We develop a character-level pretrained language model tailored for scene text, providing linguistic knowledge for short and fragmented text instances.
- We introduce a linguistic decoder initialized with our PLM, enabling tight fusion of linguistic and visual information and improving recognition of ambiguous or visually degraded text.

2. Related Work

2.1. End-to-end Scene Text Spotting

End-to-end scene text spotting [4, 7, 8, 15, 18, 21, 39, 41, 45] localizes and recognizes text from natural scene images within a single network. These type of approaches enable shared feature representations between text detection and recognition, and jointly optimizes both tasks, leading to improved performance.

The increasing success of transformer models has expanded their application to computer vision tasks. DETR-based text spotting methods [7, 8, 17, 39, 41, 45] have achieved state-of-the-art performance. TESTR [45] consists of a single encoder and dual decoders, one for predicting control points localizing text regions and the other for character recognition. DeepSolo [41] performs transformer-based text spotting with a single decoder, localizing text regions with Bézier curves. ESTextSpotter [8] enables explicit communication between the detection and recognition tasks. SwinTextSpotter v2 [9] proposes Recognition Conversion and Recognition Alignment modules to strengthen the synergy between detection and recognition. While these end-to-end text spotters demonstrate strong performance, the explicit use of prior linguistic knowledge has remained largely unexplored until recently.

2.2. Language-Aware Scene Text Spotting

Recent work in scene text spotting has explored incorporating linguistic clues to enhance performance. LMTextSpotter [39] introduces a separate language modeling recognition branch trained in an autoregressive manner to capture character-level dependencies within a text sequence. This autoregressive design aligns with modeling strategies used in natural language processing (NLP), where decoder-only transformer models such as GPT [30] generate text by predicting the next token based on previously generated tokens. Because LMTextSpotter trains only on text from scene images, it cannot fully benefit from the linguistic priors offered by broadly pretrained language models.

Existing pretrained language models (PLMs) [5, 13, 20, 30, 31] are generally trained on large-scale text corpora such as news articles and books and are designed to learn the linguistic context of tokens within well-structured sentences. TextBlockV2 [24] leverages the pretrained GPT-2 [30] to initialize the text spotting decoder. This approach extracts visual embeddings for each multi-word text block using an encoder, and then decodes these embeddings to produce block-level text recognition outputs. This design attempts to leverage the ability of PLMs to capture linguistic information across sentence-level contexts. However, because PLMs typically learn to capture complete sentences with coherent semantic meaning, they do not naturally align with scene text spotting, where text often appears as short, spatially localized words or phrases that rarely combine into a complete sentence. As a result, such block-level approaches struggle to achieve consistently strong performance on standard scene text spotting benchmarks.

2.3. Text Recognition with Language Modeling

Several works have incorporated language modeling into the scene text recognition (STR) task [1, 14, 37], which differs from text spotting in that it skips detection and assumes cropped text regions are directly provided as input.

Two general approaches exist that differ from TICLS in mechanism or decoding granularity. First, some approaches adopt language modeling objectives from NLP tasks while maintaining a character-level decoding space, including masked language modeling [37]. For example, PARSeq [1] uses permutation language modeling [40] for causal self-attention and cross-attends to visual encoder outputs with decoder-side linguistic embeddings. This approach helps STR to learn linguistic regularities implicitly. TrOCR [14] directly integrates a pretrained language model (PLM) into the text recognition task, decoding over sub-word tokens. As described in Section 2.2, integrating existing PLMs for text recognition tends to perform well on long text sequences, such as lines, phrases, or sentences (e.g., handwritten text recognition), rather than on word-level text from STR. Although recent STR work incorporates linguis-

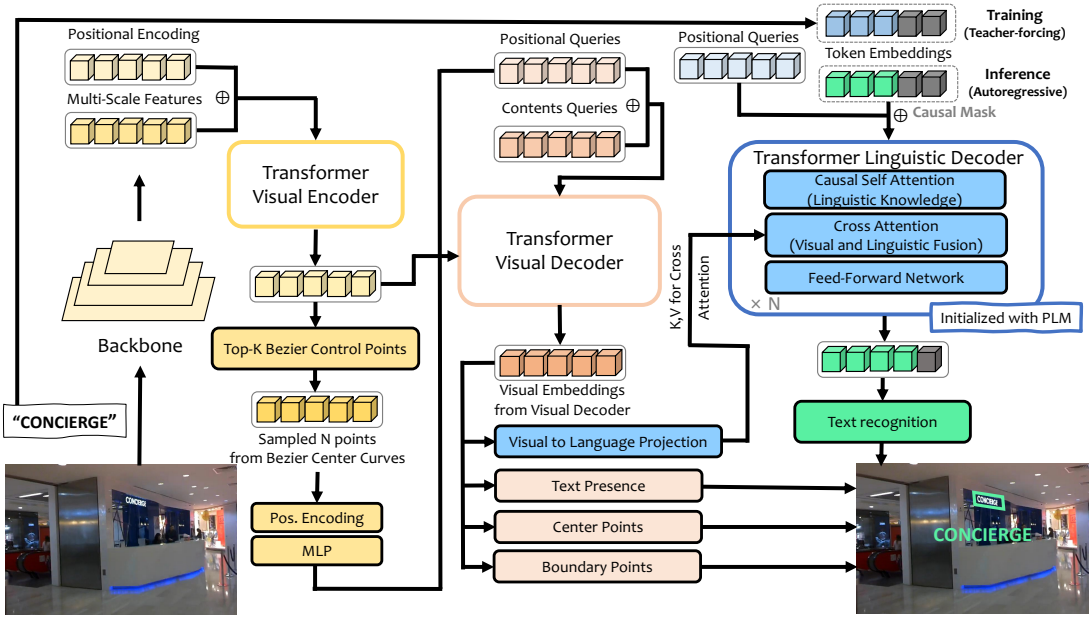

Figure 1. Overall architecture of TICLS. TICLS builds on a DETR-style visual encoder (yellow) and visual decoder (orange). A linguistic decoder (blue) receives visual representations from the visual-to-language projection head and performs visual-linguistic fusion for text recognition (green). The linguistic decoder is initialized with our proposed PLM (Section 3.2).

tic cues with notable success, integrating linguistic information into end-to-end scene text spotting remains underexplored, which forms a core contribution of this work.

3. TICLS for Text Spotting

TICLS consists of three components (Figure 1): a visual feature extractor, a visual decoder, and a linguistic decoder. For the visual components, we adapt DeepSolo [41], a DETR-based end-to-end text spotter. We remove DeepSolo's character classification head and introduce a new visual-to-language projection module to map the visual features into the input space required by the linguistic decoder. The linguistic decoder fuses the transformed visual features with prior knowledge by initializing the decoder with a pretrained language model specifically designed for scene text.

Section 3.1 reviews the visual feature extractor and decoder; Section 3.2 introduces the overall architecture and the pretraining of our proposed language model; and Section 3.3 describes the linguistic decoder in TICLS, which fine-tunes the integrated PLM decoder.

3.1. Visual Feature Extractor and Decoder

TICLS builds on DeepSolo's [41] visual feature extractor and decoder. The visual feature extractor leverages a ViTAEv2-based backbone to obtain multi-scale visual features, followed by a visual encoder. The visual encoder produces text region proposals together with their text-existence scores, and TICLS selects the top K proposals based on these scores for further processing. For each proposal, we sample N points along the Bézier curves derived from the encoder output. We apply a sinusoidal positional encoding function to these points and feed the encoded representations into an MLP to obtain the position queries for the visual decoder. In the visual decoder, we add these position queries to the learnable content queries, and the resulting representations are iteratively refined through intra- and inter-sequence self-attention, which captures relationships among character-level visual features, and through cross-attention with the visual encoder outputs.

The decoder produces $H^{\text{vis}} \in \mathbb{R}^{K \times N \times D_{\text{V}}}$ as the input to three heads: i) a binary text-presence classifier, ii) a center points head regressing centerline Bézier offsets, and iii) a boundary points head regressing top/bottom Bézier offsets.

To bridge the visual and linguistic decoders, we introduce a visual-to-language projection head that maps embeddings from the visual decoder's D_{V}-dimensional space into the linguistic decoder's D_{L}-dimensional space. The projected embeddings serve as the input to the linguistic decoder (Section 3.3), which performs text recognition with tightly coupled visual and linguistic representations.

3.2. Pretrained Language Model for Scene Text

Figure 2 shows the architecture of our proposed language model, which consists of a transformer with a bidirectional encoder and a left-to-right decoder, inspired by BART [13]. The encoder learns bidirectional context from corrupted text and the decoder predicts each token autoregressively based

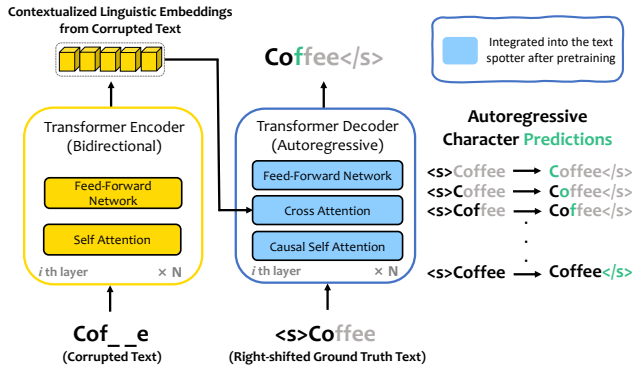

Figure 2. Proposed PLM architecture. The PLM encoder (yellow block) generates contextualized embeddings from the corrupted text, and the PLM decoder (blue block) autoregressively generates text (green) based on the encoder outputs. TICLS initializes its linguistic decoder with the PLM decoder (blue block).

on its preceding tokens and the encoder output. The core intuition is that the decoder, trained to reconstruct words from corrupted text during the PLM pretraining, enables TICLS to better handle visually corrupted embeddings.

Our approach differs from standard BART pretraining in three respects: i) Token granularity. Unlike standard BART, which uses a sub-word token vocabulary typical of PLMs, we adopt character-level tokens to better match short, fragmented scene text. ii) Input sequences. We pretrain the PLM on text sequences of at most three words instead of full sentences, which mirrors scene text characteristics. iii) Decoder cross attention source. During pretraining, the PLM decoder cross-attends to the PLM encoder states. However during the training and inference for TICLS, the decoder instead cross-attends to the projected visual features from TICLS's visual decoder.

We follow BART's denoising sequence-to-sequence pretraining paradigm. Among the multiple corruption functions explored in BART (i.e., token masking, deletion, text infilling, sentence permutation, and document rotation), we adopt text infilling, masking 20-40% of contiguous alphabetic spans while preserving non-alphabetic symbols such as digits and separators. Let $\mathbf{y} = (y_1, \ldots, y_S)$ denote a text sequence of length S, and let T be the maximum sequence length such that $S \leq T$. We generate a corrupted version of the sequence, $\tilde{\mathbf{y}} = (\tilde{y}_1, \ldots, \tilde{y}_T)$, using a text infilling corruption function with padding tokens from positions $S+1$ to T. The bidirectional transformer encoder, composed of stacked self-attention blocks, processes $\tilde{\mathbf{y}}$ to produce contextualized hidden representations $H^{\text{enc}} \in \mathbb{R}^{T \times D_{\text{L}}}$, where D_{L} denotes language model hidden dimension size. The left-to-right transformer decoder reconstructs the original sequence $\mathbf{y}$. Each decoder layer applies masked self-attention and cross-attention to the encoder states, followed by a position-wise feed-forward network and a linear prediction head.

During training, we use a teacher forcing mechanism [38] to guide autoregressive output. The decoder receives a right-shifted target sequence $\mathbf{y}^{\rightarrow} = (\texttt{<s>}, y_1, \ldots, y_{T-1})$ and predicts the next token y_t at each step based on the previous tokens $\mathbf{y}_{<t}$ and the encoder outputs H_{enc}. The decoder self-attention applies a causal mask $M \in \mathbb{R}^{T \times T}$ to prevent access to future tokens:

$$M_{pq} = \begin{cases} 0, & q \leq p \\ -\infty, & q > p \end{cases} \quad (p, q = 1, \ldots, T). \tag{1}$$

We train the model by minimizing the token-level negative log-likelihood, BART's training objective,

$$\mathcal{L}_{\text{PLM}} = -\sum_{t=1}^{T} \log p(y_t \mid \mathbf{y}_{<t},\, H^{\text{enc}}). \tag{2}$$

At inference, the model autoregressively decodes without ground truth input, generating its own predictions until the maximum length T is reached.

3.3. Linguistic Decoder

TICLS's linguistic decoder shares the PLM's decoder architecture. We initialize the linguistic decoder in TICLS with the PLM's decoder weights (Section 3.2), including all sub-layers in each decoder block (i.e., masked self-attention, cross-attention and position-wise feedforward network) and the linear prediction head. This initialization transfers linguistic knowledge to TICLS; although pretrained to predict text from embeddings of corrupted input, further training enables the linguistic decoder to recognize scene text by adapting to visual features.

Upon initialization, we switch the linguistic decoder's cross-attention source from the PLM encoder states H^{enc} to the projected visual features Z^{vl} produced by TICLS's visual decoder, while keeping the architecture unchanged. In this setup, we train the decoder on scene text data with both visual and linguistic inputs, enabling a tightly coupled integration of the two modalities in the end-to-end text spotter. Finally, the decoder states pass through the linear prediction head to produce text recognition output autoregressively.

After integrating the linguistic decoder into TICLS, we use the same teacher-forcing training strategy used in PLM pretraining. The decoder receives the right-shifted target sequence $\mathbf{y}^{\rightarrow}$ and attends to the visual-linguistic features Z^{vl}. We similarly train the model by minimizing the token-level negative log-likelihood:

$$\mathcal{L}_{\text{text}} = -\sum_{t=1}^{T} \log p\big(y_t \mid \mathbf{y}_{<t},\, Z^{\text{vl}}\big). \tag{3}$$

At inference, TICLS follows the same autoregressive decoding process as in PLM pretraining, generating tokens

sequentially until producing an end-of-sequence token or reaching the maximum length T.

3.4. Optimization and Loss

Consistent with DeepSolo [41] and previous DETR-based text spotting methods [8, 45], we find the optimal bipartite matching between the ground truth and model outputs to calculate loss. For both the visual encoder and visual decoder outputs, we use the Hungarian bipartite matching algorithm [12] to pair predictions with ground truth by minimizing a cost function.

The overall training objective of TICLS is

$$\mathcal{L} = \mathcal{L}_{\text{vis-enc}} + \mathcal{L}_{\text{vis-dec}} + \mathcal{L}_{\text{lang-dec}}. \tag{4}$$

For the visual encoder, we use $\mathcal{L}_{\text{vis-enc}}$, which consists of a classification loss $\mathcal{L}_{\text{cls}}$ based on focal loss [16] and a control point loss $\mathcal{L}_{\text{coord}}$ based on ℓ_1 distance. The corresponding loss weights are $\lambda_{\text{enc-cls}}$ and $\lambda_{\text{enc-coord}}$. For the visual decoder, we use $\mathcal{L}_{\text{vis-dec}}$, which includes $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{coord}}$, and an additional boundary point loss $\mathcal{L}_{\text{bd}}$ using ℓ_1 distance. Their respective weights are $\lambda_{\text{dec-cls}}$, $\lambda_{\text{dec-coord}}$, and $\lambda_{\text{dec-bd}}$.

Lastly, the linguistic decoder loss $\mathcal{L}_{\text{lang-dec}}$ incorporates the text classification loss $\mathcal{L}_{\text{text}}$. We use the cross-entropy loss for $\mathcal{L}_{\text{text}}$ in an autoregressive manner, as described in Eq. 3, with corresponding weighting hyperparameter λ_{text}:

$$\mathcal{L}_{\text{lang-dec}} = \sum_{k} \left(\lambda_{\text{text}} \, \mathcal{L}_{\text{text}}^{(k)} \right) \tag{5}$$

where $k \in \{1, \ldots, K\}$ denotes the index of the k-th query, and K is the number of decoder queries (i.e., top encoder proposals).

4. Experiments and Results

Here we provide a comprehensive evaluation and ablation study demonstrating how tightly coupling a pretrained language model purpose-built for scene text produces superior performance across multiple benchmarks.

4.1. Benchmarks for Scene Text Spotting

We evaluate TICLS with **ICDAR 2015** [11], **Total-Text** [2] and **CTW1500** [19]. ICDAR 2015 includes 1,000 training images and 500 testing images with word-level annotations. Total-Text focuses on curved and arbitrarily oriented word-level text, containing 1,255 training images and 300 testing images. CTW1500 differs from these datasets by emphasizing curved phrases, where each annotated unit may contain multiple words. CTW1500 includes 1,000 training images and 500 testing images.

For training, we leverage four datasets: i) **SynthText 150K** [21], featuring 149,050 synthetic images; ii) **ICDAR 2013** [10], comprised of 229 training images and 223 testing images from the ICDAR 2013 Robust Reading competition; iii) **TextOCR** [33], containing 21,778 scene images for training; and iv) **ICDAR 2017-MLT 2017** [27], a multilingual scene text spotting benchmark consisting of 7,200 training images; we use only the Latin annotations.

4.2. Datasets for PLM

To pretrain the language model for scene text, we leverage three data sources: i) **text spotting and recognition benchmarks**: We collect the text from multiple scene text training datasets including SynthText 150K [21], Total-Text [2], ICDAR 2013 [10], ICDAR 2015 [11], TextOCR [33], LSVT [34], ArT [3], SVT [36], and IIIT 5K [26]. These datasets contain text in real-world scene images, typically consisting of one or two words. We *exclude* the text in the validation and testing splits of each dataset. ii) **an NLP corpus**: We extract text from AG News [44] and WikiText [25], which are widely used in NLP pretraining tasks. iii) **Points of Interest (POI)**: We randomly sample OpenStreetMap [28] POI names in Paris, Barcelona, and Beijing, which contain POIs with Latin characters and provide diverse background and scene text styles. We select approximately 4,000 POIs for Paris, 4,000 for Barcelona, and 2,200 for Beijing. We only use POIs whose names consist exclusively of Latin characters. The entire dataset contains 932,905 unique text sequences.

4.3. Implementation and Training Details

Following DeepSolo, we allow $K = 100$ text proposals and sample $N = 25$ points per curve. The visual encoder and decoder each use 6 layers with hidden size $D_{\text{V}} = 256$. Like BART our linguistic decoder uses hidden size $D_{\text{L}} = 768$. The visual-to-language projection head uses a two-layer MLP to map visual to linguistic embeddings $\mathbb{R}^{D_{\text{V}}} \rightarrow \mathbb{R}^{D_{\text{L}}}$. Loss weights are $\lambda_{\text{coord}} = \lambda_{\text{bd}} = \lambda_{\text{cls}} = 1$ and $\lambda_{\text{text}} = 6$ throughout the experiments, except that $\lambda_{\text{text}} = 15$ when fine-tuning TICLS on CTW1500.

We pretrain TICLS on four NVIDIA A100-SXM4-40GB GPUs with a batch size of 4. For fine-tuning on each target benchmark, we train TICLS with a batch size of 2 on two GPUs (due to resource constraints). We initialize TICLS's backbone, visual encoder and decoder weights from DeepSolo's pretrained weights. For training TICLS, we adopt a step learning rate schedule that starts at 10^{-4} and decays to 10^{-5} at 240K iterations and use AdamW for optimization. We then fine-tune TICLS using the training set in the evaluation benchmarks (i.e., either ICDAR 2015, Total-Text, or CTW1500). During fine-tuning, we adopt a step learning rate schedule that starts at 10^{-5} and decays to 10^{-6} at 45K iterations for Total-Text, and a fixed learning rate of 10^{-6} for ICDAR 2015. For CTW1500, we use a step learning rate schedule that starts at 10^{-5}, decays to 10^{-6} at 10K iterations, and further decays to 10^{-7} at 20K iterations.

Compared to existing text spotting methods that predict 96 characters, our model predicts 194 tokens, including

Method	Year	ICDAR 2015				Total-Text		CTW1500	
		Strong	Weak	Generic	None	None	Full	None	Full
ABCNet [21]	2020	82.7	78.5	73.0	–	70.4	78.1	45.2	74.1
Mask TextSpotter v3 [15]	2020	83.3	78.1	74.2	–	71.2	78.4	–	–
MANGO [29]	2021	81.8	78.9	67.3	–	72.9	83.6	58.9	78.7
ABCNet v2 [22]	2021	83.0	80.7	75.0	–	73.5	80.7	57.5	77.2
TESTR [45]	2022	85.2	79.4	73.6	65.3	73.3	83.9	56.0	81.5
GLASS [32]	2022	84.7	80.1	76.3	–	79.9	86.2	–	–
DeepSolo [41]	2023	88.1	83.9	79.5	73.8	83.6	89.6	64.2	81.4
ESTextSpotter [8]	2023	87.5	83.0	78.1	–	80.8	87.1	66.0	83.6
SPTS v2 [23]	2023	81.2	74.3	68.0	–	75.0	82.6	63.6	84.3
FastTCM [42]	2024	87.0	82.0	77.3	–	79.9	87.2	60.4	78.8
IATS [43]	2024	84.4	80.0	73.8	64.7	71.9	83.5	62.4	82.9
FastTextSpotter [4]	2024	86.6	81.6	75.4	–	75.1	86.0	56.0	82.9
LMTextspotter [39]	2024	87.5	83.4	78.4	–	81.1	88.4	62.5	80.1
TextBlockV2 [24]	2025	–	–	80.5	–	80.0	–	65.1	83.3
SwinTextSpotter v2 [9]	2025	89.6	84.1	79.4	–	82.8	88.4	61.3	82.0
TICLS (ours)	2026	**90.1**	**85.4**	**81.9**	**79.1**	**84.6**	**90.4**	**66.2**	**88.1**

Table 1. End-to-end text spotting results on ICDAR 2015, Total-Text, and SCUT-CTW1500. The standard F1-measure (Hmean) is reported.

punctuation, symbols, and case-sensitive characters. In particular, for characters that appear at the beginning of a word except for the first word in the sequence, we use distinct token variants to reflect their word-initial position, following standard practices in NLP tokenization [5, 13, 20]. For example, in the phrase "apple pie", the **p** in "a**p**ple" and the **p** in "**p**ie" are treated as distinct tokens to reflect their word-initial positions. The PLM matches the linguistic decoder's hidden size D_{L} across its 6 encoder layers and 6 decoder layers. A feed-forward network (FFN) module also follows the standard Transformer architecture, which includes a dimensionality expansion from $D_{\mathrm{L}} = 768$ to 3072, followed by a contraction back to D_{L}, with GELU activation and layer normalization. We pretrain the language model for 40 epochs with two GPUs and a batch size of 128.

4.4. Results and Discussion

Table 1 presents evaluation results on ICDAR 2015, Total-Text, and CTW1500, comparing TICLS with existing state-of-the-art end-to-end text spotting methods. Overall, our method achieves the best performance across all datasets and standard evaluation settings.

ICDAR 2015. We evaluate under three standard lexicon settings: i) Strong (S) provides a lexicon containing the words annotated in each image. ii) Weak (W) uses the union of lexicons from all evaluation images. iii) Generic (G) contains additional words beyond the ground truth set. As a post-processing step, like TESTR and its DETR-based descendants [7–9, 41, 45], we use an absolute edit distance threshold of 2 when selecting a lexicon entry closest to the predicted output, discarding any detections exceeding the threshold. TICLS surpasses all prior baseline methods, improving over SwinTextSpotter v2 by 0.5% on Strong and 1.3% on Weak, and over TextBlockV2 by 1.4% on Generic. In addition, TICLS achieves a lexicon-free Hmean of 79.1%, which shows that adding the linguistic

(a) (b)

Figure 3. Qualitative comparison on Total-Text (a and b) between TICLS (first row) and DeepSolo (second row). Green (ours) and yellow (DeepSolo) indicate correct detection and recognition results, while red (DeepSolo) denotes incorrect recognition results.

decoder boosts performance by 5.3% over DeepSolo.

Total-Text. We evaluate under two standard settings: without a lexicon (None) and with a lexicon constructed from all evaluation text instances (Full), applying the same edit distance-based filtering and correction. Compared to existing methods, TICLS attains the best performance in both settings, with improvements over 1% on None and 0.8% on Full compared to DeepSolo.

CTW1500. We evaluate TICLS on CTW1500 to assess its ability to handle multi-word scene text without a lexicon (None) and with a "lexicon" containing the full text phrase instances from the test set. Because of the longer text sequences in this dataset compared to others, we adopt a normalized edit-distance threshold of 0.3 for post-processing. TICLS achieves the best performance in both settings, yielding a 0.2% improvement over ESTextspotter in the None setting and a 3.8% improvement over SPTSv2 when using the lexicon.

Qualitative Analysis. Figure 3 provides a qualitative comparison between TICLS and DeepSolo, the state-of-the-art model on Total-Text. In Figure 3(a), our model correctly predicts the text *LOSTWORLD* (9 characters), while DeepSolo mis-recognizes the first character, producing *TOSTWORLD* by confusing the visually similar "L" and "T". Similarly, in Figure 3(b), the word *ASHBURY* appears rotated against a visually complex background. DeepSolo incorrectly recognizes the last character as "E", whereas TICLS accurately predicts the correct word. These examples highlight the robustness of our tightly integrated model under diverse scene text conditions.

4.5. Detailed Analysis

To assess the effect of incorporating explicit linguistic information, we further analyze performance by examining word-level recognition accuracy with respect to text sequence length and vocabulary coverage, comparing our method with DeepSolo. Since our primary contribution lies in the recognition branch, we report accuracy only for word instances that both models detect correctly.

Text Sequence Length. We group the ground-truth text into three bins based on the number of characters in each sequence. Table 2 shows the results for ICDAR 2015. This factoring allows us to quantify how well each model handles short fragments compared to longer, more complex sequences. Both models achieve relatively high accuracy on short sequences (3–5 characters), with our model slightly outperforming the baseline. The performance gap widens for medium-length sequences (6–10 characters), where our model improves performance nearly 10%. The most significant difference appears for longer sequences of 11 or more characters, where our model excels by over 28%. These results demonstrate that TICLS's capability of incorporating external linguistic information for scene text spotting offers significant benefits for long text sequences, particularly in the lexicon-free setting.

Word Length (chars)	Model	Correct Detections (Intersection)	Correct Recognitions	Accuracy (%)
3-5	DeepSolo	1055	933	88.4
	TICLS	1055	953	90.3
6-10	DeepSolo	697	533	76.4
	TICLS	697	601	86.2
11+	DeepSolo	39	17	43.5
	TICLS	39	28	71.7

Table 2. ICDAR 2015 (None) word recognition accuracy for various word lengths.

Model	PLM Pre-Training Text	OOV	Instances	Accuracy (%)	
				Lexical	Overall
DeepSolo	–	No	1322	89.8	84.1
		Yes	368	63.5	
TICLS	Scene Text	No	1322	93.8	88.4
		Yes	368	68.7	
TICLS	External Text	No	1466	92.2	87.7
		Yes	224	58.0	
TICLS	External Text ∪ Scene Text	No	1466	93.3	89.3
		Yes	224	62.9	

Table 3. Comparison of word recognition accuracy between words seen during training and out-of-vocabulary (OOV) test words on ICDAR 2015 (None). External Text is the NLP and POI data.

Vocabulary Coverage. Inspired by the out-of-vocabulary (OOV) text spotting challenge [6], we evaluate the effect of expanding the training vocabulary. Table 3 summarizes the results. The first row reports the word-level accuracy of DeepSolo. The second and third rows present the performance of TICLS initialized with PLMs trained on different corpora: one using *only* text from scene text spotting datasets, and the other using *only* external text data (i.e., the NLP and POI datasets of Section 4.2). The last row corresponds to our proposed setting, where we train the PLM with *both* scene text spotting datasets and external text. Note that OOV counts are identical in the last two rows because fully training TICLS involves the same scene text data, regardless of how the PLM was pretrained.

Across all methods, DeepSolo and the three variants of TICLS consistently show higher accuracy on in-vocabulary words than out-of-vocabulary, regardless of the pretraining text. Comparing the first two rows reveals that integrating the PLM into the text spotter improves word recognition accuracy even without introducing additional vocabu-

PLM Initialization	Vocab. size	Granularity	S	W	G	None
Our PLM (Section 3.2)	194	character	90.1	85.4	81.9	79.1
BART-base	50 265	subword	83.3	77.2	73.1	69.0

Table 4. End-to-end F1-score results (Hmean) on ICDAR 2015 comparing the linguistic decoder initializations.

Methods	Vocabulary size	None
DeepSolo	96	73.8
DeepSolo	194	75.5
TiCLS (ours)	194	79.1

Table 5. End-to-end F1-score results (Hmean) on ICDAR 2015 comparing vocabulary size.

lary, yielding gains of 4.0% for in-vocabulary words and 5.2% for out-of-vocabulary words. Comparing the third and fourth rows further shows that learning linguistic prior knowledge from scene text during PLM pretraining is essential, as reflected by the 4.9% boost in out-of-vocabulary performance. Moreover, in overall accuracy, our proposed PLM configuration achieves an additional 0.9% and 1.6% improvement over the versions trained only on scene text and only on external text, respectively, and a net 5.2% improvement over DeepSolo. Finally, we observe that training TiCLS without any PLM pretraining (e.g. a randomly initialized linguistic decoder) fails to converge.

4.6. Ablation Studies

This section presents ablation experiments to evaluate the effectiveness of a key design choice in TiCLS: token granularity. We first compare an existing PLM with a subword-level token design to our character-level token design. We then compare our token design with DeepSolo's 96-character vocabulary, widely used in existing text spotters. We conduct the ablation on the ICDAR 2015 test set for consistency with experiments in Section 4.4.

PLM Initialization. To evaluate the effectiveness of our proposed PLM design for scene text spotting, we compare two initializations of the linguistic decoder: our proposed character-level PLM and a strong off-the-shelf language model. In particular, we consider the BART decoder [13] pretrained on sentence-level inputs with a 50K-token subword vocabulary. We then fully train the text spotter from each initialization, as described in Section 4.3.

Table 4 shows that our PLM, with a compact vocabulary of 194 character-level tokens, consistently outperforms the BART-initialized variant. Across different lexicon settings, our model achieves substantial gains. The margin narrows under stronger lexicons, suggesting that lexicon guidance partially compensates for the limitations of sub-word representations by providing external linguistic constraints. The character-level formulation could be more robust to truncated or irregular inputs common in scene text, whereas subword-based models may suffer from unstable segmentations, especially when dealing with very short phrases consisting of only one or two words.

In addition, given the 50K-token vocabulary, fine-tuning the BART-initialized variant is inherently challenging, which could also account for the weaker performance of the subword-based model. Overall, the results demonstrate that our proposed character-level PLM provides an effective and robust initialization for scene text spotting, yielding consistent improvements over subword-based alternatives.

Token vocabulary. To isolate the contribution of our linguistic decoder, we train DeepSolo using the same 194-class vocabulary as TiCLS. We replace the original 96-class character classification head of DeepSolo with a 194-class head and train with the same settings used to train TiCLS. The results in Table 5 show that even with a modified vocabulary, DeepSolo's performance falls well short of TiCLS's, highlighting the effectiveness of the proposed linguistic decoder beyond simply enlarging the token set.

5. Conclusion, Limitations, and Future Work

We introduce TiCLS, an end-to-end scene text spotter that tightly incorporates external linguistic knowledge from a character-level PLM. By explicitly leveraging pretrained linguistic knowledge and integrating it with visual features, TiCLS enables robust performance across diverse and challenging cases of scene text. Our experiments show that TiCLS achieves state-of-the-art performance, demonstrating the effectiveness of tightly unified visual-linguistic modeling for scene text spotting.

Despite these performance gains, there are potential improvements to the model. Our model's parameter count is $2.73\times$ DeepSolo's (98.5M versus 36.0M); as a result the inference forward pass takes $1.55\times$ longer (1.35s versus 0.87s). This increase reflects the overhead of our design, which incorporates an additional linguistic decoder. These observations point to opportunities for refinement. One possible direction is to reduce redundancy by incorporating linguistic pretraining directly into the visual decoder. In addition, although the current inference stage in the linguistic decoder adopts greedy decoding, incorporating a beam search could further improve accuracy, especially on long sequences, at the cost of increased computation.

Acknowledgment

This material is based upon work supported in part by the National Science Foundation (Award No. 2419334) and by the University of Minnesota's Researcher Assistance Initiative for Supporting Emergencies (RAISE).

References

[1] Darwin Bautista and Rowel Atienza. Scene text recognition with permuted autoregressive sequence models. In European conference on computer vision, pages 178–196. Springer, 2022. 2

[2] 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), pages 935–942. IEEE, 2017. 5

[3] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. ICDAR2019 robust reading challenge on arbitrary-shaped text - RRC-ArT. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1571–1576, 2019. 5

[4] Alloy Das, Sanket Biswas, Umapada Pal, Josep Lladós, and Saumik Bhattacharya. FastTextSpotter: A high-efficiency transformer for multilingual scene text spotting. In International Conference on Pattern Recognition, pages 135–150. Springer, 2024. 2, 6

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT (1), pages 4171–4186, 2019. 1, 2, 6

[6] Sergi Garcia-Bordils, Andrés Mafla, Ali Furkan Biten, Oren Nuriel, Aviad Aberdam, Shai Mazor, Ron Litman, and Dimosthenis Karatzas. Out-of-vocabulary challenge report. In Computer Vision – ECCV 2022 Workshops, pages 359–375, Cham, 2023. Springer Nature Switzerland. 7

[7] 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

[8] Mingxin Huang, Jiaxin Zhang, Dezhi Peng, Hao Lu, Can Huang, Yuliang Liu, Xiang Bai, and Lianwen Jin. Estextspotter: Towards better scene text spotting with explicit synergy in transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 19495–19505, 2023. 2, 5, 6

[9] Mingxin Huang, Dezhi Peng, Hongliang Li, Zhenghao Peng, Chongyu Liu, Dahua Lin, Yuliang Liu, Xiang Bai, and Lianwen Jin. SwinTextSpotter v2: Towards better synergy for scene text spotting. International Journal of Computer Vision, pages 1–21, 2025. 2, 6

[10] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazàn Almazàn, and Lluís Pere de las Heras. ICDAR 2013 robust reading competition. In 2013 12th International Conference on Document Analysis and Recognition, pages 1484–1493, 2013. 5

[11] Dimosthenis Karatzas, Lluis Gomez Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. ICDAR 2015 competition on robust reading. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR), pages 1156–1160, 2015. 5

[12] Harold W Kuhn. The Hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97, 1955. 5

[13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019. 1, 2, 3, 6, 8

[14] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-based optical character recognition with pre-trained models. In Proceedings of the AAAI conference on artificial intelligence, pages 13094–13102, 2023. 2

[15] 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

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 5

[17] Yijun Lin and Yao-Yi Chiang. Hyper-local deformable transformers for text spotting on historical maps. In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 5387–5397, 2024. 2

[18] 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

[19] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recognition, 90:337–345, 2019. 5

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa:

A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692, 2019. 1, 2, 6

[21] 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, 5, 6

[22] 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, 2021. 6

[23] Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, and Lianwen Jin. Spts v2: Single-point scene text spotting, 2023. 6

[24] Jiahao Lyu, Jin Wei, Gangyan Zeng, Zeng Li, Enze Xie, Wei Wang, Can Ma, and Yu Zhou. TextBlockV2: Towards precise-detection-free scene text spotting with pre-trained language model. ACM Transactions on Multimedia Computing, Communications and Applications, 2025. 1, 2, 6

[25] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. 5

[26] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In BMVC, 2012. 5

[27] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, Wafa Khlif, Muhammad Muzzamil Luqman, Jean-Christophe Burie, Cheng-lin Liu, and Jean-Marc Ogier. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification - RRC-MLT. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pages 1454–1459, 2017. 5

[28] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org, 2017. 5

[29] Liang Qiao, Ying Chen, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Mango: A mask attention guided one-stage scene text spotter, 2021. 6

[30] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. OpenAI blog, 1 (8):9, 2019. 1, 2

[31] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67, 2020. 1, 2

[32] Roi Ronen, Shahar Tsiper, Oron Anschel, 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

[33] 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 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8798–8808, 2021. 5

[34] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. Icdar 2019 competition on large-scale street view text with partial labeling – rrc-lsvt, 2019. 5

[35] George Tom, Minesh Mathew, Ajoy Mondal, Jerod Weinman, Dimosthenis Karatzas, and C. V. Jawahar. Occluded road text - robust reading competition, 2024. https://rrc.cvc.uab.es/?ch=29. 1

[36] Kai Wang and Serge Belongie. Word spotting in the wild. In European conference on computer vision, pages 591–604. Springer, 2010. 5

[37] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 14194–14203, 2021. 2

[38] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. Neural computation, 1(2):270–280, 1989. 4

[39] Xin Xia, Guodong Ding, and Siyuan Li. LMTextSpotter: Towards better scene text spotting with language modeling in transformer. In International Conference on Document Analysis and Recognition, pages 76–92. Springer, 2024. 1, 2, 6

[40] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32, 2019. 2

[41] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. DeepSolo: Let transformer decoder with explicit points solo for text spotting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 19348–19357, 2023. 2, 3, 5, 6

[42] Wenwen Yu, Yuliang Liu, Xingkui Zhu, Haoyu Cao, Xing Sun, and Xiang Bai. Turning a clip model into a scene text spotter, 2023. 6

[43] Shi-Xue Zhang, Chun Yang, Xiaobin Zhu, Hongyang Zhou, Hongfa Wang, and Xu-Cheng Yin. Inverse-like antagonistic scene text spotting via reading-order estimation and dynamic sampling, 2024. 6

[44] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification, 2016. 5

[45] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. pages 9519–9528, 2022. 2, 5, 6