COO: Comic Onomatopoeia Dataset for Recognizing Arbitrary or Truncated Texts

Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa

The University of Tokyo
{baek,matsui,aizawa}@hal.t.u-tokyo.ac.jp

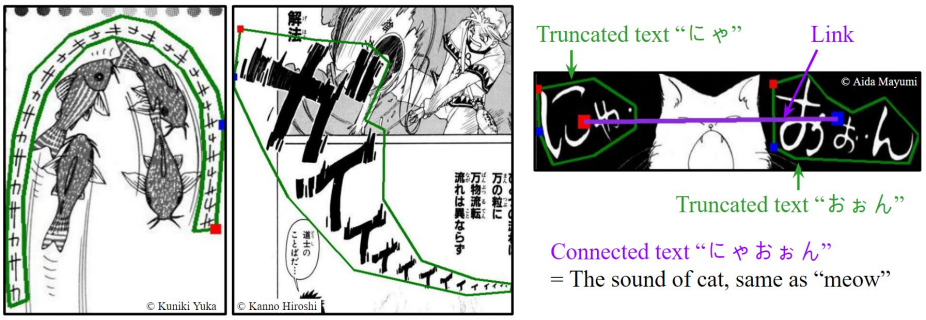
Abstract. Recognizing irregular texts has been a challenging topic in text recognition. To encourage research on this topic, we provide a novel comic onomatopoeia dataset (COO), which consists of onomatopoeia texts in Japanese comics. COO has many arbitrary texts, such as extremely curved, partially shrunk texts, or arbitrarily placed texts. Furthermore, some texts are separated into several parts. Each part is a truncated text and is not meaningful by itself. These parts should be linked to represent the intended meaning. Thus, we propose a novel task that predicts the link between truncated texts. We conduct three tasks to detect the onomatopoeia region and capture its intended meaning: text detection, text recognition, and link prediction. Through extensive experiments, we analyze the characteristics of the COO. Our data and code are available at https://github.com/ku21fan/COO-Comic-Onomatopoeia.

Keywords: Comic Onomatopoeia, Arbitrary Text, Truncated Text, Text Detection, Text Recognition, Link Prediction

1 Introduction

Along with the development of deep neural networks, text recognition methods have significantly improved. Currently, most state-of-the-art methods can easily recognize simple horizontal texts. Recently, the research trend has progressed to recognize more irregular texts: recognizing horizontal text to recognizing arbitrary-shaped text such as curved or perspective text in scene images [10, 20, 22, 23, 25, 42–45, 49, 51]. We expect that *studies on more irregular texts will further improve the text recognition methods.*

To encourage these studies, we provide a novel comic onomatopoeia dataset (COO), which contains more irregular texts. After investigating various text datasets from English [9,15–17,30,36,40,50] to other languages [1,8,11,26,27,34, 37,48], we find that onomatopoeia texts in the Japanese comic dataset (Manga10-9 [26]) have arbitrary shapes or are arbitrarily placed in the image. Onomatopoeias are written texts that represent the sound or state of objects (humans, animals, and so on). To exaggerate the sound or state of the object, onomatopoeias are typically written in irregular shapes or placed at unexpected positions. Fig. 1 (a) illustrates examples of COO: (left) shows extremely curved text, (right) shows partially shrunk text, and part of the text is on the object.



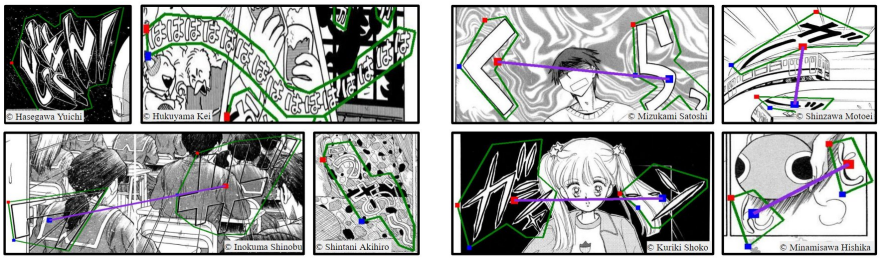
(a) Arbitrary texts (b) Truncated texts and link between them

Fig. 1. Visualization of comic onomatopoeia dataset *COO*. Red and blue squares denote the start and end points of each annotation, respectively. The purple line denotes the link between truncated texts

Onomatopoeia in Japanese comics is sometimes separated into several parts, as shown in Fig. 1 (b). When separated, each part is a truncated text. Each truncated text does not fully represent the meaning. After truncated texts are connected, the connected text represents the intended meaning. For example, the truncated texts “にゃ” and “おぁん” in Fig. 1 (b) do not represent the meaning independently. When they are connected into “にゃおぁん”, the connected text represents the meaning: the sound of cat, same as “meow”. To correctly capture the meaning of truncated texts, *we propose a novel task that predicts the link between truncated texts*. By using the link information, we connect truncated texts to capture the intended meaning. To solve this task, we formulate the task as the sequence-to-sequence problem [38], and propose a model named M4C-COO, a variant of multimodal multi-copy mesh (M4C) [13].

Considering truncated texts, we conduct three tasks to detect the onomatopoeia region and capture its intended meaning: 1) Text detection: The model takes an image, and outputs the regions of onomatopoeias. 2) Text recognition: The model takes the region of onomatopoeia, and outputs the text in the region. 3) Link prediction: The model takes the regions and texts of onomatopoeias, and outputs the links between truncated texts. With extensive experiments using state-of-the-art methods, we analyze the characteristics of COO and the limitation of current models. We hope that these analyses inspire and encourage future work on recognizing arbitrary or truncated texts.

Among three tasks, we mainly focus on text recognition and link prediction. Because they are somewhat different from existing tasks, they can hinder using our dataset. To prevent it, we provide decent baselines for them. Traditional text recognition task generally recognizes horizontal or curved texts. However, the COO has many vertically long texts: In 72.5% of onomatopoeia regions, the height is greater than the width. To address vertically long texts, we introduce several effective techniques. In the case of the link prediction task, it is a novel task, and thus we introduce it thoroughly.



(a) Various onomatopoeias (b) Truncated texts and link between them

Fig. 2. Visualizations of COO. Each example shows diversity of onomatopoeias

In summary, our main contributions are as follows:

- We construct a novel challenging dataset *COO* to encourage the research on recognizing arbitrary or truncated texts.
- COO has many vertically long texts, and we investigate several techniques that are effective to recognize vertically long texts.
- COO has some truncated texts, and they should be linked. We propose a novel task, which predicts the link between truncated texts, and a M4C-COO model for this task.

2 COO: Comic Onomatopoeia Dataset

Most onomatopoeias in COO are arbitrary-shaped or arbitrarily placed. In addition, they are written in informal fonts or various sizes. This section introduces the visualization, annotation guideline, statistics, and analysis of our dataset. More details are presented in the supplementary materials.

2.1 Why Use Onomatopoeias of Japanese Comics?

We use onomatopoeias of Japanese comics rather than English comics. Because of following three reasons:

1. Japanese comics have various types of onomatopoeias. Fig. 1 and Fig. 2 show many arbitrary or truncated texts. As arbitrary texts, Fig. 2 (a) (top row) shows three-dimensional or curved texts. Fig. 2 (a) (bottom row) shows transparent texts on the objects that look similar to background objects. Fig. 2 (b) shows truncated texts.
2. Japanese comics have more onomatopoeias than English comics. We compared Japanese comic dataset Manga109 [26], and English comic dataset COMICS [15]. Manga109 has *5.8 onomatopoeias per page on average*, whereas COMICS has much fewer (492 onomatopoeias in the first 5,000 images).
3. Japanese language has more diverse onomatopoeias than most languages. According to Petersen [28], "The reason sounds in manga are so rich and varied is also in part due to the nature of the Japanese language that has a much wider range of onomatopoeic expressions than most languages."

2.2 Label Annotation

For each comic onomatopoeia, we create annotation data according to three tasks: 1) Text detection: Annotate polygon regions. 2) Text recognition: Annotate texts. 3) Link prediction: Annotate links between truncated texts.

We annotate comic onomatopoeias in Manga109 [26]. Manga109 consists of 109 Japanese comics. Each image in Manga109 consists of two pages (left and right pages) because some objects or onomatopoeias lie across two pages, as shown in Fig. 3.

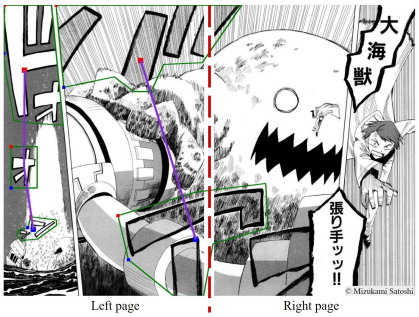


Fig. 3. Each image in Manga109 consists of two pages, and it is used as input data for text detection task

Polygon regions of onomatopoeia. We use polygon annotations instead of bounding box annotations to minimize regions irrelevant with texts. We place the points that represent the contour of the onomatopoeia. The points are placed clockwise starting from the top left of the onomatopoeia and ending with the bottom left. Red and blue squares in Fig. 2 denote start and end points of each annotation, respectively. Because most state-of-the-art methods are developed with single-line annotations, we split a multi-lined onomatopoeia into single lines as much as possible.

Texts of onomatopoeia. While the Japanese language consists of Hiragana (e.g. “あ”, “ぃ”), Katakana (e.g. “ア”, “イ”), and Chinese characters (e.g. “任”, “意”), Japanese comic onomatopoeias are typically written in Hiragana or Katakana. Thus, Chinese characters are not the target of annotation. We annotate Hiragana, Katakana, and some special symbols for each onomatopoeia. Generally, most comic onomatopoeias are written in informal fonts or by drawing.

Link between truncated texts. Link annotation is conducted on polygon and text annotations. The link between truncated texts is determined by the meaning of onomatopoeia. Onomatopoeias have a link if the following conditions are satisfied: 1) Two or more onomatopoeias are separated in the image. 2) By themselves, they do not fully represent the intended meaning. 3) When they are connected, they represent the intended meaning. The purple line in Fig. 2 denotes the link between truncated texts.

The annotation was performed with an annotation team consisting of 15 annotators and 3 annotation checkers. After all onomatopoeias in 109 comics were annotated, annotation checkers performed the initial check. After that, we (authors) checked the annotations over three times and provided feedback for re-annotation to ensure annotation quality. As a result, annotations have been revised over three times.

2.3 Dataset Analysis

Table 1 presents the statistics of COO dataset. COO has 61,465 polygons in total. If we regard the polygon that has more than 4 points as curved, the ratios of the curved, quadrilateral, and rectangular annotations are 61.3%, 15.4%, and 23.4%, respectively. The average number of points on all polygons is 6.3. COO has 2,261 links in total, and one link appears every five pages on average. Most links are between two truncated texts. The numbers of links made by three, four, and five truncated texts are 132, 11, and 1, respectively.

Table 1. Statistics on the COO dataset

Count type	Total	Train	Valid	Test
Images	10,602	8,763	890	949
Comic volumes	109	89	10	10
Polygon	61,465	50,064	4,636	6,765
Link	2,261	1,923	161	177
Vocabularies	13,272	11,635	1,915	2,251
Character types	182	182	163	166

The number of character types is 182, and Fig. 4 shows all of them. To recognize English texts, one may use 94 characters, including alphanumerics and symbols, as done in ASTER [33]. To recognize general Japanese texts, one should use thousands of Chinese characters. Then, the number of character types exceeds thousands, and the increment of the character types makes text recognition difficult. However, comic onomatopoeias do not include thousands of Chinese characters, and character types do not increment considerably, only 94 to 182. This indicates that the difficulty from the increment of the character types is little, and we can focus on recognizing arbitrary or truncated texts.

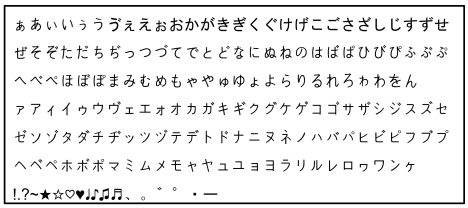


Fig. 4. Total 182 character types of COO. Because COO does not contain Chinese characters, the number of character types is much smaller than that in the Japanese language (over 2,000 characters)

COO can be also used for comic analysis or comic translation. For example, 79% of links start from the left and end with the right. This is an interesting characteristic because Japanese comics generally read right to left while the order of most truncated texts is reverse. For other example, in our manual check over each truncated texts, we find that an object is typically placed between truncated texts. This is assumed to be a drawing technique to represent the sound or state of the object dramatically.

2.4 Comparison with Existing Arbitrary Scene Text Datasets

Comic onomatopoeias exhibit various shapes and sizes, and are placed arbitrarily in the image. They are close to scene text rather than document text. There are several existing arbitrary-shaped scene text datasets: CUTE [30], CTW1500 [21],

Fig. 5. Examples of truncated texts in English

Total-Text [9], ArT [8], and TextOCR [36]. They have polygon annotations for curved texts. While these datasets focus on arbitrary-shaped texts, our dataset COO focuses on both arbitrary-shaped texts and arbitrarily placed texts. Furthermore, while most texts in other datasets are horizontal or curved, and are not separated into several parts, our dataset has many vertical texts or texts separated into several parts (truncated texts). Thus, we mainly focus on vertical text recognition and link prediction between truncated texts.

Our dataset contains only Japanese comic onomatopoeias whereas other datasets mainly contain English or Chinese texts. One may concern that methods developed on our dataset may not be generalized to other cases. However, we believe that the algorithm developed on Japanese comic onomatopoeias can be generalized to other cases because the algorithm developed on the small English benchmark data was generalized to other languages. Scene text detection and recognition methods have been developed based on English datasets. According to the benchmark paper of scene text recognition [3], the total number of English benchmark evaluation data is 8,539. The number of character types and vocabulary of the data are 79 and 3,940, respectively. The data definitely do not cover all the texts in real life. However, the developed algorithm for competing on this small English benchmark data did not differ from the winning algorithms in ICDAR2019 competitions to recognize multi-lingual texts [27] and Chinese texts [8, 37, 48].

2.5 Truncated Texts in English

Truncated texts (onomatopoeias) in Japanese comics and link prediction for connecting them might be considered too special. However, it is not, and similar problems also occur in other cases. Fig. 5[1] (left) shows that the words on necklace are separated into two pieces: "BEST" → "BE" and "ST", and "FRIENDS" → "FRIE" and "NDS". (middle) shows that the word "Summer" is separated into "S" and "ummer". (right) shows that the word "SUMMER" is separated into "SUM" and "MER". Like the (right) image, we sometimes see a word separated in multiline in the poster or commercials. In general, current state-of-the-art methods are specialized in single-line recognition, not multiline. Here, we can use link prediction to connect them ("SUM" and "MER") and capture the intended

[1] The images in Fig. 5 can be found here: left, middle, right (accessed 03-08-2022)

meaning ("SUMMER"). Extending the link prediction to these cases can be a new problem of our community.

3 Methods for Three Tasks

We summarize our methods for three tasks. Text detection and recognition are well-known tasks; thus, we skip details of model description. Meanwhile, since link prediction between truncated texts is a novel task proposed in this study, we thoroughly introduce the M4C-COO model for the task. More details are in the supplementary materials.

3.1 Text Detection

Text detection methods are mainly categorized into regression-based [10, 22, 25, 42, 45, 49, 51] and segmentation-based methods [20, 23, 43, 44]. To investigate the appropriate approach for comic onomatopoeia, we use two methods in each category. Specifically, we use ABCNet v2 [22] and MTS v3 [20] as representatives of regression-based and segmentation-based methods, respectively. Both methods were originally proposed for the text spotting task in which text detection and recognition tasks are combined. However, these methods also provided results of using only the text detection part and showed state-of-the-art performance. We take the only text detection part and use them as text detectors. Furthermore, MTS v3 exhibits superior performance for rotated text detection. We expect that MTS v3 can also detect vertical texts in our dataset.

3.2 Text Recognition

In this study, the well-known model called TPS-ResNet-BiLSTM-Attention (TRBA) [3] is used. TRBA is created by combining existing methods such as RARE [32] and FAN [7]. TRBA takes four steps to recognize texts: 1) Rectify input image with TPS transformation [6]. 2) Convert rectified images into visual features by ResNet [12]. 3) Convert visual features into contextual features by BiLSTM. 4) From contextual features, predict character string with attention module [4].

3.3 Link Prediction

In this study, we formulate the link prediction task into the sequence-to-sequence problem [38]. The model takes the sequence of all onomatopoeias in an image, and outputs the sequence of truncated texts. The sequence of truncated texts consists of pairs of truncated texts and the delimiter symbol <d> which divides pairs of truncated texts. Under this setting, predicting links between truncated texts is the same as predicting the sequence of truncated texts from the input sequence.

An example of input and output sequences is as follows. Given the input sequence as below and truncated texts are (1) "ド" and "ッ" separated from "ドッ", and (2) "ボ" and "ン" separated from "ボン" (when two pairs of truncated texts exist), the output sequence is as follows.

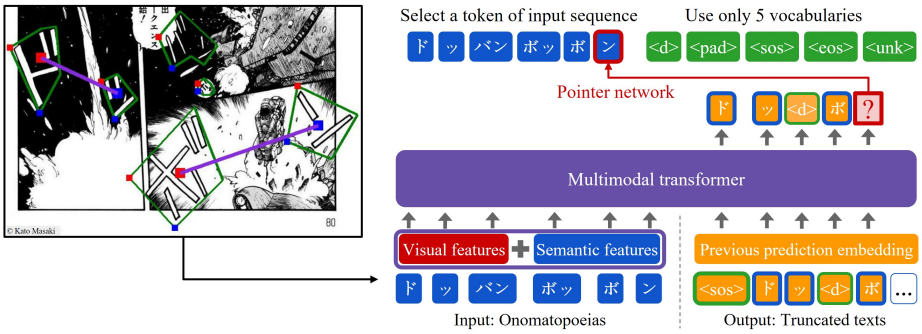


Fig. 6. M4C-COO takes the sequence of all onomatopoeias in an image and outputs the sequence of truncated texts

Input sequence: [ド, ッ, バン, ボッ, ボ, ン]
Output sequence: [ド, ッ, <d>, ボ, ン]

By dividing the output sequence with <d>, we obtain two lists [ド, ッ] and [ボ, ン]. By connecting each of the lists, we obtain connected texts "ドッ" and "ボン". Fig. 6 illustrates this example.

To solve this sequence-to-sequence problem, we propose a model named M4C-COO. Fig. 6 illustrates M4C-COO. M4C-COO is a variant of a model called multimodal multi-copy mesh (M4C) [13]. M4C has been used for visual question answering with text (TextVQA [35,36]). M4C takes question word embedding, object, and OCR (text) tokens and fuses them using a multimodal transformer [39]. In addition, M4C uses an iterative answer prediction mechanism to generate a multi-word answer. Based on fused features and iterative answer prediction, M4C predicts the answer of the question. Unlike M4C for TextVQA [13], M4C-COO does not use question word embedding and object part. M4C-COO takes only onomatopoeia tokens and predicts a sequence of truncated texts.

To find truncated texts, we should exploit both visual and semantic (text) features because 1) truncated texts look similar and are close, and 2) They represent the intended meaning if they are connected. M4C-COO exploits both visual and semantic features of onomatopoeias. In M4C-COO, onomatopoeia tokens are embedded into four features, which are categorized as visual and semantic features. For visual features, we use (1) appearance feature (FRCN) from onomatopoeia regions extracted by using Faster RCNN [29] part in MTS v3 [20], and (2) 4-dimensional relative bounding box coordinates (bbox) for each onomatopoeia region. For semantic features, we use (3) fastText [5] and (4) pyramidal histogram of characters (PHOC) [2]. fastText is a word embedding method with sub-word information. fastText is well known for handling out-of-vocabulary words. PHOC counts characters in the word and makes the pyramidal histograms for each word.

Furthermore, the copy mechanism of the pointer network [41] in M4C-COO is exactly what we needed for link prediction. Generally, the copy mechanism selects a token (word) in the input sequence, and the selected token is used as

an output token. In other words, it copies the token in the input sequence to the output sequence. In our task, we need to select truncated texts in the input onomatopoeia sequence. This operation is the same as the copy mechanism.

M4C generally uses both thousands of vocabularies and copy mechanism to predict the output sequence. Thousands of vocabularies are used to generate a token that is not in the input sequence. In our task, all the tokens in the output sentence are in input sentence, except for the delimiter symbol <d> and end of sentence token <eos>. Thus, *M4C-COO uses only five vocabularies*: the delimiter symbol <d> and four special tokens for training M4C-COO, padding token <pad>, start and end of sentence tokens (<sos> and <eos>), and unknown token <unk>.

4 Experiment and Analysis

In this section, we present the results of the experiments on three tasks. Through experiments, we analyze the characteristics of COO and the limitations of the current methods. More details of the experimental settings are provided in the supplementary materials.

4.1 Implementation Detail

Model and training strategy. For text detection, we use the official codes of ABCNet v2[2] [22] and MTS v3[3] [20]. For text recognition and link prediction, we use the official codes of TRBA[4] [3] and M4C[5] [13], respectively. For the training strategy, we follow the default setting to the extent possible.

Dataset. We split 109 comics of Manga109 into 89, 10, and 10 books and use them as training, validation, and test sets, respectively. For the evaluation, we select the model with the best score on the validation set. In each task, we use ground truth information rather than predicted results of other tasks.

Evaluation metric. For text detection, we use intersection over union to determine whether the model correctly detects the region of onomatopoeia. For text detection and link prediction, we show precision (P), recall (R), and their harmonic mean (H, Hmean). As a default, we mainly use Hmean for comparison. For text recognition, we show word-level accuracy for comparison. We run three trials for all experiments and report average values.

4.2 Text Detection

We compare the effectiveness of bounding box annotation and polygon annotation, and compare the regression-based detector with the segmentation-based detector.

[2] https://github.com/aim-uofa/AdelaiDet/tree/master/configs/BAText
[3] https://github.com/MhLiao/MaskTextSpotterV3
[4] https://github.com/clovaai/deep-text-recognition-benchmark
[5] https://github.com/facebookresearch/mmf/tree/main/projects/m4c

Table 2. Ablation study on text detectors ABCNet v2 and MTS v3

#	Method	P	R	H
1	ABCNet v2 [22]-Bounding box	61.9	60.7	61.2
2	ABCNet v2 [22]-Polygon	67.7	64.5	66.0
3	MTS v3 [20]-Bounding box	67.5	58.2	62.5
4	MTS v3 [20]-Polygon	**69.8**	**65.9**	**67.8**

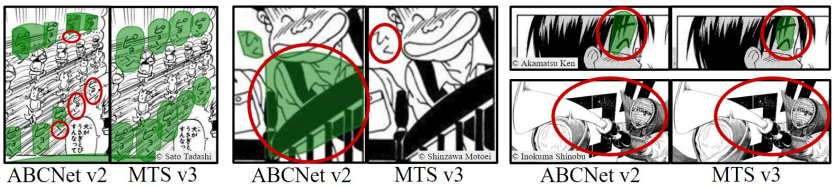


Fig. 7. Text detection on the test set. The green regions are the predicted regions and the red circles are failures

Training with bounding box vs. with polygon. Lines 1 and 3 in Table 2 show the results of training with bounding box (the axis-aligned rectangle that bounds the onomatopoeia region) annotation instead of using polygon annotation. Comparing lines (1 vs. 2) and (3 vs. 4), training with polygon annotation shows better performance than training with bounding box annotation: +4.8% for ABCNet v2 and +5.3% for MTS v3. However, the performance improvement by using polygon annotation is not that considerable. This result may indicate that current detection algorithms may not fully exploit polygon annotation. To improve performance, we may need the algorithm that exploits irregular polygon annotations, such as partially shrunk polygon, more effectively.

Regression vs. segmentation. ABCNet v2 and MTS v3 are the representatives of regression-based and segmentation-based methods, respectively. Comparing lines 2 and 4 in Table 2, MTS v3 shows better performance +1.8% than ABCNet v2. This result is unexpected and interesting because ABCNet v2 shows better performance than MTS v3 in other benchmark datasets such as MSRA-TD500 [47] (85.2 vs. 83.5). This result indicates that segmentation-based methods can be advantageous for detecting the region of onomatopoeia.

Visualization and failure case. Fig. 7 shows the predictions on the test set and failure cases of the current methods. In Fig. 7 (left), MTS v3 correctly detects vertically long onomatopoeias (nine of "ピョン"), whereas ABCNet v2 misses two onomatopoeias ("ピョン") and the part of onomatopoeias ("ン"). According to MTS v3 paper [20], MTS v3 is good at detecting rotated texts. These cases show that MTS v3 can also be used for vertically long texts. In Fig. 7 (middle), MTS v3 sometimes misses small onomatopoeias (the size of onomatopoeia "ひく" is $[width \times height] = [25 \times 28]$ whereas the image size is

Table 3. Ablation study on text recognizer TRBA

#	Method	Accuracy
1	TRBA [3]	46.3
2	+ Rotation trick	49.8
3	+ SAR decoding	43.2
4	+ Rotation trick + SAR decoding	55.4
5	#4 + Hard RoI (batch 100%)	63.5
6	#4 + Hard RoI (batch 50%)	67.9
7	#6 + 2D attention (height 64)	78.5
8	#6 + 2D attention (height 100)	**81.0**

[1654×1170]) and ABCNet v2 misclassifies the text-like part (similar to "ン") as onomatopoeia. Both methods misclassify the text-like part (similar to "サヘ") as onomatopoeia (right-top) and miss the occluded texts (right-bottom).

4.3 Text Recognition

We investigate techniques to address vertically long texts, such as rotation and decoding tricks, Hard RoI masking, and 2D attention.

Rotation and decoding tricks. A text recognizer generally takes a text image in which characters are arranged horizontally as input and recognizes each character from left to right. However, if the model takes a text image in which characters are arranged vertically as input, the model cannot recognize each character from left to right. For this case, a simple rotation trick can be useful: Rotates vertical images 90 degrees to make them horizontal. Here, an image whose height is greater than the width and whose text label contains more than two characters is regarded as a vertical image. Comparing lines 1 to 2 in Table 3, using the rotation trick results in a performance gain of +3.5%.

Some cases, such as short vertical texts, are correctly recognized without the rotation trick but incorrectly recognized with the rotation trick. For these cases, SAR decoding [19] can be useful. SAR decoding is a decoding trick of the text recognition model SAR [19]: At the test time, if the height of the input image is greater than the width, the model takes three images as input data: the original image and images rotated by -90 and 90 degrees. The confidence score on recognizing each image is calculated. Next, the model outputs the result with the highest confidence score. Lines 3 and 4 in Table 3 show that solely adding SAR decoding results in a performance drop of -3.1%, whereas using both the rotation trick and SAR decoding improves the original TRBA by +9.1%.

Hard RoI masking. When the text is diagonally long, the image contains more background noise, as shown in Fig. 8 (a) (left). Background noise may be irrelevant to text and decrease performance. To suppress this region, we use the hard region of interest (Hard RoI) masking [20] that removes this region,

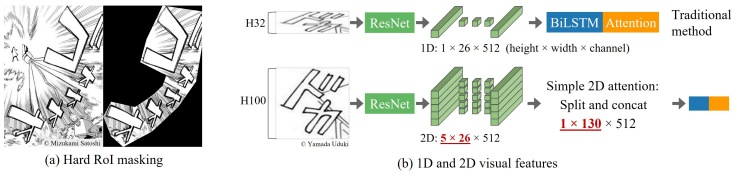


Fig. 8. (*a*) Hard RoI masking removes the region irrelevant to text. (*b*) Traditional methods use 1D attention on 1D features whereas 2D attention exploits 2D features

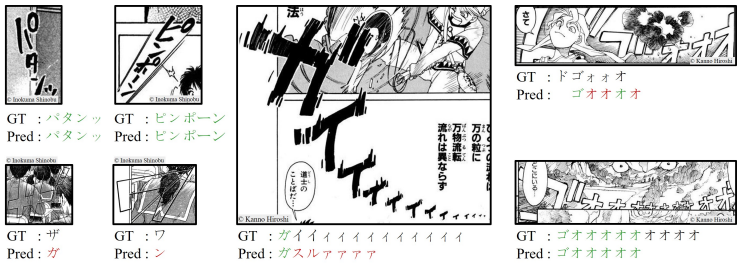


Fig. 9. Text recognition on the test set. GT denotes the ground truth, and Pred denotes the prediction by TRBA with the best score (#8 in Table 3). Green- and red-colored characters denote correct and incorrect recognition, respectively

as shown in Fig. 8 (a) (right). Comparing lines 4 and 5 in Table 3, using Hard RoI masking shows an improvement by +8.1%. Furthermore, considering that evaluation is conducted without Hard RoI masking, teaching the model how to handle original images can be useful. To do so, we fill half of each mini-batch with original images. As shown in line 6, performance further improves by +4.4%.

2D attention on 2D visual features. The model can benefit from considering the attention on vertical direction. Traditional methods [3,7,31–33] take an image with a height of 32 and make 1D visual features through convolutional networks (ResNet), as shown in Fig. 8 (b) (top): $[height \times width \times channel] = [1 \times 26 \times 512]$ is called 1D because the vertical dimension is squeezed to 1. Because English texts are mainly horizontal, most methods follow this trend. Recently, some methods [14, 18, 19, 24, 46] take an image whose height is greater than 32 and make 2D visual features, as shown in Fig. 8 (b) (bottom): $[5 \times 26 \times 512]$ is called 2D because the vertical dimension remains at *5*. These methods showed performance improvements by using 2D attention on 2D visual features.

We show the effectiveness of 2D attention with a minimal modification. We use a simple method to exploit 2D visual features, as shown in Fig. 8 (b) (bottom). We use 1D BiLSTM, and we need to make vertical dimension 1 before BiLSTM. To do so, we split vertical features and concatenate them horizontally (e.g., 5×26 to 1×130), as done in EPAN [14]. Lines 7 and 8 in Table 3 show performance improvements. Simple 2D attention improves performance by +10.6% and +13.1% where the heights of images are 64 and 100, respectively.

Table 4. Ablation study on link prediction model M4C-COO

#	Method	Visual feature	Semantic feature	P	R	H
1	Rule-base	distance	–	1.1	**74.5**	2.1
2	M4C-COO	FRCN + bbox	fastText + PHOC	**77.2**	68.7	**72.7**
3	+ Vocab. 11,640	FRCN + bbox	fastText + PHOC	55.0	38.7	45.4
4	Only fastText	–	fastText	42.4	30.2	35.2
5	+ PHOC	–	fastText + PHOC	61.7	50.5	55.4
6	+ PHOC + FRCN	FRCN	fastText + PHOC	62.1	53.4	57.2

Visualization and failure case. Some vertical or diagonal texts can be correctly recognized, whereas transparent, partially shrunk, or occluded texts are incorrectly recognized. Fig. 9 shows the predictions by TRBA with the best score (#8 in Table 3): TRBA correctly recognizes vertically long texts "パタンッ" and "ピンポ—ン" (row 1, column 1 and 2) whereas incorrectly recognizes transparent texts "ザ" and "ワ" (row 2, column 1 and 2), partially shrunk text (column 3), and texts occluded by objects or a frame (column 4).

4.4 Link Prediction

We present the results of the link prediction task, such as a comparison with a rule-based method and ablation studies.

M4C-COO vs. distance-based method. We test the distance-based method as a baseline: 1) Calculate the average distance from one truncated text to another truncated text. 2) For each onomatopoeia, if the other onomatopoeia is closer than the average distance, they are regarded as linked. Line 1 in Table 4 shows that the distance-based method has the highest recall of 74.5% but a considerably low precision of 1.1%. Comparing lines 1 and 2, M4C-COO shows a much better performance of +70.6% (Hmean) than the distance-based method.

Effect of vocabulary. Comparing lines 2 and 3 in Table 4, *M4C-COO (with only five vocabularies) shows a much better performance of +27.3% than M4C-COO with 11,640 vocabularies (vocabularies of the training set)*. This result indicates that using only the copy mechanism is more suitable for this task than using both many vocabularies and the copy mechanism. Many vocabularies may disturb the copy mechanism, and therefore performance decreases.

Ablation study. Line 4 in Table 4 shows that using only fastText feature for M4C-COO results in a performance drop of -37.5%. Line 5 shows that if we use PHOC together, the performance drastically improves by +20.2%. Line 6 indicates that if FRCN (appearance feature) is added, the performance further improves by +1.8%. However, the performance gain by adding FRCN is considerably less than that by adding PHOC. This result is reasonable considering the drawing style. Because the comic artist is identical in each image, the drawing

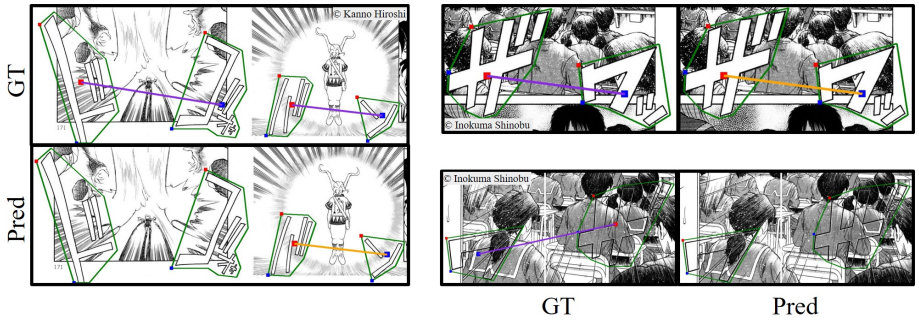


Fig. 10. Link prediction on the test set. The purple line denotes the ground truth link between truncated texts, and the orange line denotes the predicted link

(writing) style of onomatopoeias in each image is similar. Therefore, exploiting the appearance feature to predict the link is less effective. Comparing lines (2 vs. 6), using bbox (relative coordinate) makes huge improvement by +15.5%. Sometimes there are multiple onomatopoeias whose texts are identical in an image, and only one of them is a truncated text linked to other truncated text. In such cases, the model can benefit from bbox. The model can select the only one truncated text by using coordinate.

Visualization and failure case. Some links can be correctly predicted, whereas predicting links between texts similar to background images is difficult. Fig. 10 shows the predictions by M4C-COO: M4C-COO correctly predicts the link between "バ" and "ン" (column 1) and the link between "ザ" and "ワッ" (column 2, row 1), while misses the link between "ド" and "ゴォォ" (column 1) and the link between transparent texts "ザ" and "ワ" (column 2, row 2).

5 Conclusion

We have constructed a novel dataset named COO. COO contains many arbitrary-shaped texts or arbitrarily placed texts. Some texts are separated into several parts, and each part is a truncated text. To capture the intended meaning of truncated texts, we have proposed the link prediction task and the M4C-COO model. We have conducted three tasks (text detection, text recognition, and link prediction) and provided decent baselines. We have experimentally analyzed the characteristics of COO and the limitation of current methods. Detecting the onomatopoeia region and capturing the intended meaning of truncated texts are not straightforward. Thus, COO is a challenging text dataset. We hope that this work will encourage future work on recognizing various types of texts.

Acknowledgements. This work is partially supported by JSPS KAKENHI Grant Number 22J13427, JST-Mirai JPMJMI21H1 and AI center of the University of Tokyo.

References

1. Aizawa, K., Fujimoto, A., Otsubo, A., Ogawa, T., Matsui, Y., Tsubota, K., Ikuta, H.: Building a manga dataset "manga109" with annotations for multimedia applications. IEEE MultiMedia (2020) 1
2. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. TPAMI (2014) 8
3. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: ICCV (2019) 6, 7, 9, 11, 12
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR (2015) 7
5. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. TACL (2017) 8
6. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. TPAMI (1989) 7
7. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: Towards accurate text recognition in natural images. In: ICCV (2017) 7, 12
8. Chng, C.K., Liu, Y., Sun, Y., Ng, C.C., Luo, C., Ni, Z., Fang, C., Zhang, S., Han, J., Ding, E., et al.: Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In: ICDAR (2019) 1, 6
9. Ch'ng, C.K., Chan, C.S., Liu, C.: Total-text: Towards orientation robustness in scene text detection. IJDAR (2020) 1, 6
10. Dai, P., Zhang, S., Zhang, H., Cao, X.: Progressive contour regression for arbitrary-shape scene text detection. In: CVPR (2021) 1, 7
11. Guérin, C., Rigaud, C., Mercier, A., Ammar-Boudjelal, F., Bertet, K., Bouju, A., Burie, J.C., Louis, G., Ogier, J.M., Revel, A.: ebdtheque: a representative database of comics. In: ICDAR (2013) 1
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 7
13. Hu, R., Singh, A., Darrell, T., Rohrbach, M.: Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In: CVPR (2020) 2, 8, 9
14. Huang, Y., Sun, Z., Jin, L., Luo, C.: Epan: Effective parts attention network for scene text recognition. Neurocomputing (2020) 12
15. Iyyer, M., Manjunatha, V., Guha, A., Vyas, Y., Boyd-Graber, J., Daume, H., Davis, L.S.: The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In: CVPR (2017) 1, 3
16. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: ICDAR (2015) 1
17. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: ICDAR (2013) 1
18. Lee, J., Park, S., Baek, J., Oh, S.J., Kim, S., Lee, H.: On recognizing texts of arbitrary shapes with 2d self-attention. In: Workshop on Text and Documents in the Deep Learning Era, CVPR (2020) 12
19. Li, H., Wang, P., Shen, C., Zhang, G.: Show, attend and read: A simple and strong baseline for irregular text recognition. In: AAAI (2019) 11, 12
20. Liao, M., Pang, G., Huang, J., Hassner, T., Bai, X.: Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In: ECCV (2020) 1, 7, 8, 9, 10, 11

21. Liu, Y., Jin, L., Zhang, S., Luo, C., Zhang, S.: Curved scene text detection via transverse and longitudinal sequence connection. Pattern Recognition (2019) 5
22. Liu, Y., Shen, C., Jin, L., He, T., Chen, P., Liu, C., Chen, H.: Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. TPAMI (2021) 1, 7, 9, 10
23. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C.: Textsnake: A flexible representation for detecting text of arbitrary shapes. In: ECCV (2018) 1, 7
24. Lu, N., Yu, W., Qi, X., Chen, Y., Gong, P., Xiao, R., Bai, X.: Master: Multi-aspect non-local network for scene text recognition. Pattern Recognition **117**, 107980 (2021) 12
25. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. TMM (2018) 1, 7
26. Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., Aizawa, K.: Sketch-based manga retrieval using manga109 dataset. MTAP (2017) 1, 3, 4
27. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al.: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In: ICDAR (2019) 1, 6
28. Petersen, R.S.: The acoustics of manga. In Jeet Heer and Kent Worcester (Eds.) A Comics Studies Reader (pp.163-171). University Press of Mississippi (2009) 3
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS (2015) 8
30. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. ESWA (2014) 1, 5
31. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. TPAMI (2016) 12
32. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: CVPR (2016) 7, 12
33. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: Aster: An attentional scene text recognizer with flexible rectification. TPAMI (2018) 5, 12
34. Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., Belongie, S., Lu, S., Bai, X.: Icdar2017 competition on reading chinese text in the wild (rctw-17). In: ICDAR (2017) 1
35. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: CVPR (2019) 8
36. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: CVPR (2021) 1, 6, 8
37. Sun, Y., Ni, Z., Chng, C.K., Liu, Y., Luo, C., Ng, C.C., Han, J., Ding, E., Liu, J., Karatzas, D., et al.: Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In: ICDAR (2019) 1, 6
38. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NeurIPS (2014) 2, 7
39. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017) 8
40. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv:1601.07140 (2016) 1
41. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: NeurIPS (2015) 8

42. Wang, H., Lu, P., Zhang, H., Yang, M., Bai, X., Xu, Y., He, M., Wang, Y., Liu, W.: All you need is boundary: Toward arbitrary-shaped text spotting. In: AAAI (2020) 1, 7
43. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: CVPR (2019) 1, 7
44. Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu, G., Shen, C.: Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In: ICCV (2019) 1, 7
45. Wang, Y., Xie, H., Zha, Z.J., Xing, M., Fu, Z., Zhang, Y.: Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In: CVPR (2020) 1, 7
46. Yang, X., He, D., Zhou, Z., Kifer, D., Giles, C.L.: Learning to read irregular text with attention mechanisms. In: IJCAI (2017) 12
47. Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: CVPR (2012) 10
48. Zhang, R., Zhou, Y., Jiang, Q., Song, Q., Li, N., Zhou, K., Wang, L., Wang, D., Liao, M., Yang, M., et al.: Icdar 2019 robust reading challenge on reading chinese text on signboard. In: ICDAR (2019) 1, 6
49. Zhang, S.X., Zhu, X., Hou, J.B., Liu, C., Yang, C., Wang, H., Yin, X.C.: Deep relational reasoning graph network for arbitrary shape text detection. In: CVPR (2020) 1, 7
50. Zhang, Y., Gueguen, L., Zharkov, I., Zhang, P., Seifert, K., Kadlec, B.: Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In: Scene Understanding Workshop, CVPR (2017) 1
51. Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., Zhang, W.: Fourier contour embedding for arbitrary-shaped text detection. In: CVPR (2021) 1, 7