Visual Coherence Loss for Coherent and Visually Grounded Story Generation

Xudong Hong[124], **Vera Demberg**[24], **Asad Sayeed**[3], **Qiankun Zheng**[24] **and Bernt Schiele**[14]

[1]Dept. of Computer Vision and Machine Learning, MPI Informatics
[2]Dept. of Language Science and Technology and Dept. of Computer Science, Saarland University
[3]Dept. of Philosophy, Linguistics, and Theory of Science, University of Gothenburg
[4]Saarland Informatics Campus, Saarbrücken
{xhong,vera,qzheng}@coli.uni-saarland.de
schiele@mpi-inf.mpg.de, asad.sayeed@gu.se

Abstract

Local coherence is essential for long-form text generation models. We identify two important aspects of local coherence within the visual storytelling task: (1) the model needs to represent re-occurrences of characters within the image sequence in order to mention them correctly in the story; (2) character representations should enable us to find instances of the same characters and distinguish different characters. In this paper, we propose a loss function inspired by a linguistic theory of coherence for self-supervised learning for image sequence representations. We further propose combining features from an object and a face detector to construct stronger character features. To evaluate input-output relevance that current reference-based metrics don't measure, we propose a character matching metric to check whether the models generate referring expressions correctly for characters in input image sequences. Experiments on a visual story generation dataset show that our proposed features and loss function are effective for generating more coherent and visually grounded stories.

1 Introduction

Stories play an important role in natural language understanding and generation because they are the key to the human understanding of the world (Piper et al., 2021). Automatically generating a coherent and interesting story is a complex task requiring various capabilities such as language processing, event comprehension, and world knowledge. In this paper, we focus on the visual storytelling task of generating stories from visual narratives, i.e., a sequence of images with a plot (Huang et al., 2016). Stories, unlike image captions, contain several characters and events involving recurrent characters and their interactions with each other and the environment. Especially, *characters* are among the most important aspects of story writing (Goldfarb-Tarrant et al., 2020).

However, the current state-of-the-art (SOTA) visual storytelling models often fail to generate correct referring expressions for characters (Modi and Parde, 2019). To confirm this, we generate 50 stories using a SOTA model (TAPM; Yu et al., 2021) and annotate different types of errors[1]. The most prevalent error that appears in 60% of the generated stories is *lacking local coherence* which manifests as the characters appearing in the image not being mentioned correctly in the stories. For instance, in the first two rows of Figure 1, the character Jeremy appears across all images. The SOTA model TAPM first mentions him as *Jeremy* in the first sentence but misses him in the second sentence and inconsistently misnames him as *Adam* in the last sentence.

We identify two main causes of this problem: 1) Most previous models don't represent the recurrence of characters in the image sequence explicitly. In particular, it is not clear whether the parameters in the model can capture character recurrence such that the *visual coherence*, local coherence in image sequences, can be reflected in textual coherence by repeated mentions of the recurrent characters in the story. We argue that a model that captures and represents the visual coherence better would also lead to more coherent stories.

2) Previous models mostly consider visual features without a focus on human characters, such as features extracted with a general vision backbone model. These features lack enough power to represent properties such as age, gender, face, or body shape. As a result, models using these features cannot distinguish between different instances of the same human character, which introduces wrong character mentions in generated stories.

Our **contributions** for tackling these limitations:
(a) To tackle limitation 1), we propose a new ranking loss function, *Visual Coherence Loss (VCL)*, which is inspired by models of cohesion in language (see section 4.1). VCL punishes stand-

[1] see Appendix A

Findings of the Association for Computational Linguistics: ACL 2023, pages 9456–9470
July 9-14, 2023 ©2023 Association for Computational Linguistics

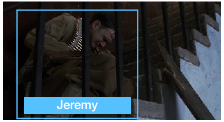
Model	 	CM	METEOR
TAPM (SOTA)	`Jeremy was hiding behind the wall . ??? Adam was able to get up and leave the building . Adam was able to escape .`	0.5	22.67
VCL	`Jeremy was in the middle of a fight with a gun . ??? Jeremy was hiding behind the wall . Jeremy is still in the room and he is trying to get out .`	0.83	38.6
VCL + Insightface	`Jeremy was hiding behind the wall . Jeremy and Adam were hiding behind the wall . Jeremy was hiding behind a wall .`	1	22.74
Oracle	`Jeremy was the one who was hiding in the room . ??? Adam was not happy with the plan . Jeremy was hiding behind the stairs .`	0.83	27.16
Human	`Jeremy was trying to escape with a gun and bullets around his neck. Adam found Jeremy sitting on the stairs. Jeremy started weeping and begging for his forgiveness.`	1	-

Figure 1: Case study of generated/human-written stories given an example image sequence. ??? denotes the error of Too Few REs defined in 5.2. The Wrong Character error is marked with a red background. We also report the CM scores together with METEOR scores.

alone instances and rewards instances that appear consecutively in visual narratives. Experiments show that the proposed loss function increases the coherence of the generated stories (see section 5).

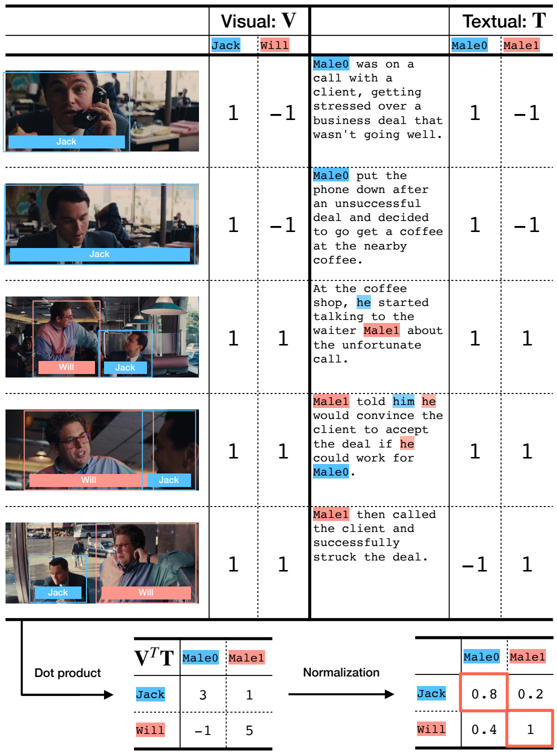
(b) To obtain better character representations and tackle limitation 2), we experiment with different features that have proven useful for person re-identification tasks (see section 6). We then add the resulting features separately to visual story generation models. Experiments show that the representation from the face detector is most effective for character representation.

(c) When evaluating the stories generated by the models, we find that reference-based metrics cannot capture referring expression errors, so we propose a new evaluation metric *character matching (CM)* for image-text relevance in visual stories (see section 3). This metric is based on a weighted bipartite matching graph between the instances in the images and the co-referring expressions in the stories. We compute a coefficient from the weighted bipartite matching that measures local coherence. We apply our metric to prove the effectiveness of our loss function.

2 Related Work

Visually-grounded Story Generation. Most of the previous work on visual story generation is based on the Visual Storytelling dataset (VIST, Huang et al., 2016). They only use global features, full image features extracted with a general vision backbone model trained on the image classification task (Yu et al., 2017; Wang et al., 2018; Huang et al., 2019). Some recent researches use local features, which are features of a specific part of the image such as objects, to generate visually grounded stories (Wang et al., 2020; Hong et al., 2020; Yu et al., 2021; Qi et al., 2021; Braude et al., 2022). Only a few works make use of human-related features like emotions (Li et al., 2019), sentiments (Chen et al., 2022) or persona (Chandu et al., 2019). One work that focuses on recurrences of characters is Dong et al. (2019), which employs a Bi-LSTM to encode conference chains in texts. However, their model requires textual features from the corresponding conference chains as input in the test time. On the contrary, our model only requires visual features from the images. Parallel to our research, Liu and Keller (2023) annotate recurring characters in the VIST dataset which enable further exploration of grounding human characters.

Coherent Story Generation. Coherence is one of the major properties that people would like to achieve in story generation. Previous work generates a story conditioned on a prompt such as keywords (Yao et al., 2019), story plot (Rashkin et al., 2020) or cue phrases (Xu et al., 2020). Training data for story generation includes the STORIUM dataset (Akoury et al., 2020). Its collaboratively-written long stories contain rich annotations such as narrator prompts, character goals, and other attributes to guide story generation. Previous models based on these datasets require access to character labels which are expensive to obtain and not always available in real-world settings. In contrast, our model only needs an additional loss function



	Visual: **V**			Textual: **T**	
	Jack	Will		Male0	Male1
	1	−1	Male0 was on a call with a client, getting stressed over a business deal that wasn't going well.	1	−1
	1	−1	Male0 put the phone down after an unsuccessful deal and decided to go get a coffee at the nearby coffee.	1	−1
	1	1	At the coffee shop, he started talking to the waiter Male1 about the unfortunate call.	1	1
	1	1	Male1 told him he would convince the client to accept the deal if he could work for Male0.	1	1
	1	1	Male1 then called the client and successfully struck the deal.	−1	1

$\mathbf{V}^T\mathbf{T}$	Male0	Male1
Jack	3	1
Will	−1	5

	Male0	Male1
Jack	0.8	0.2
Will	0.4	1

Figure 2: Computation process of character matching metric demonstrated with an example pair of image sequence and text. We show the corresponding Visual appearance matrices ($\mathbf{V}$) and textual appearance matrices ($\mathbf{T}$) of characters. The resulting maximum weighted bipartite match is in red boxes.

during model training.

Character Representations for NLG. Characters are essential to stories written and read by humans. Unfortunately, only a limited amount of work has been done on constructing explicit character representations. In story generation, Clark et al. (2018) represent characters with separate trainable parameters and integrate them with the latent representations of the language decoder using max pooling. But they only apply their method to referring expression generation instead of end-to-end story generation. In movie description, Rohrbach et al. (2017) extract head and body representations for characters and train the model for character re-identification and movie description jointly. However, all these methods integrate character features into RNN language decoders. Our model is the first attempt to encode character explicitly in Transformer-based models.

Model	**LF**	**CM**	**std**
Lower bound	-	32.04	-
Seq2Seq		68.03	1.27
Seq2Seq	obj	67.41	0.88
TAPM		67.33	1.13
TAPM	obj	67.99	0.97
Human	-	73.65	-
Upper bound	-	100	-

Table 1: Comparison of baseline models using different local features (LF) on the test set of VWP using character matching metric (CM). All numbers of models are an average of three runs with different random seeds.

3 Image-Text Relevance Metric

In previous visually-grounded story generation work, models are evaluated with reference-based metrics like METEOR. However, the reference-based metrics only measure similarities between reference texts and generated stories such that the actual relevance of input images to output text is not evaluated. Specifically, we design a character-matching metric (CM) to determine whether expressions referring to human characters in image sequences are generated correctly.

The computation process of CM is in Figure 2. For each pair of image sequence and story, we first construct the appearance matrices $\mathbf{V} \in \{1, -1\}^{m \times n_v}$ and $\mathbf{T} \in \{1, -1\}^{m \times n_t}$, which indicate for each character whether it is present (1) vs. absent (-1) in an image or a sentence, where n_v is the number of characters in the image sequence, n_t is the number of referring expressions in generated stories and m is the number of images/texts. We then compute the matrix of matching scores of each pair as: $\mathbf{M} = \mathbf{V}^T\mathbf{T}$. We normalize the matching scores to the domain $[0, 1]$ and obtain the matrix $\mathbf{M}' = (\mathbf{M}/m + \mathbf{1})/2$. To obtain the optimal match between the images and sentences by the occurrences of characters, we compute the maximum weighted bipartite matching by applying the Hopcroft-Karp algorithm (Hopcroft and Karp, 1973) implemented in SciPy (Virtanen et al., 2020). The final overall character matching score is obtained by averaging the scores of the maximum matches across all characters (e.g. 0.9 in Figure 2).

3.1 Analysis of Baselines

We apply our CM metric to show the limitation of current models, which requires the labels of characters in image sequences. We use Visual Writing

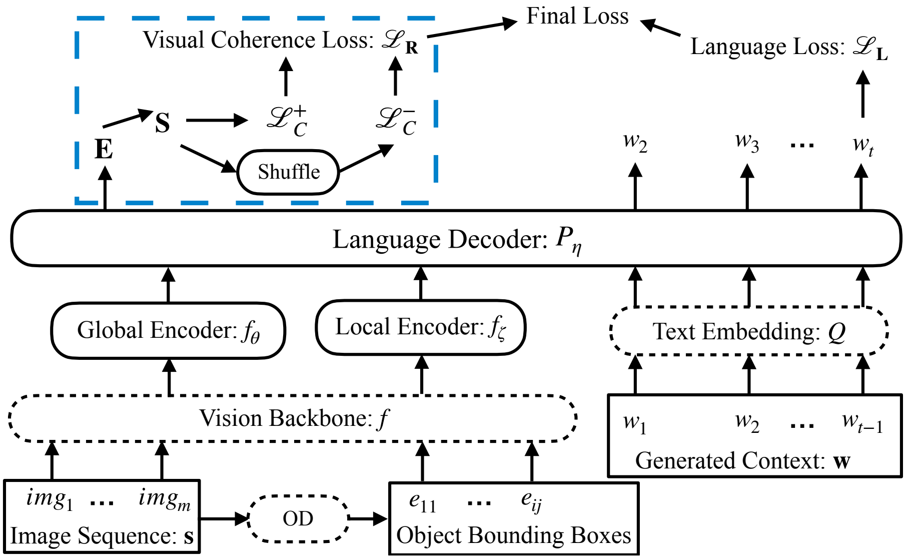

Final Loss
Visual Coherence Loss: $\mathscr{L}_\mathbf{R}$
Language Loss: $\mathscr{L}_\mathbf{L}$
$\mathbf{S}$
$\mathscr{L}_C^+$
$\mathscr{L}_C^-$
$\mathbf{E}$
Shuffle
w_2
w_3 ...
w_t
Language Decoder: P_η
Global Encoder: f_θ
Local Encoder: f_ζ
Text Embedding: Q
Vision Backbone: f
w_1 w_2 ... w_{t-1}
Generated Context: $\mathbf{w}$
img_1 ... img_m
Image Sequence: $\mathbf{s}$
OD
e_{11} ... e_{ij}
Object Bounding Boxes

Figure 3: Architecture of visual story generation model with visual coherence loss. Our contributions are in the **blue box**. Components in dash lines are frozen during training. OD is the pre-trained object detector.

Prompts (VWP; Hong et al., 2023), a vision-based dataset that contains image sequences aligned with human-written stories in English[2]. In VWP, each image is corresponding to a piece of text. Instances of each main character are annotated with the name of the character (see Figure 1). Our baselines are:

Seq2Seq. (Seq2Seq; Huang et al., 2016) is a seq2seq model with an encoder-decoder architecture. Visual features are first projected with an encoder which is a feed-forward layer, then fed to the decoder which is a pre-trained GPT-2 model.

TAPM. (TAPM; Yu et al., 2021) is a Transformer-based model which adapts the visual features with pre-trained GPT-2. This is the current state-of-the-art model for visual story generation.

For generated text, we run a coreference resolution model (Lee et al., 2018) to capture all referring expressions. We compute the CM metric for the generated stories of two baseline models as well as the human-written stories. We compute the CM upper bound (100) by treating gold annotations of characters in image sequences as referring expressions in texts, i.e. using $\mathbf{V}$ as textual appearance matrix. We also estimate the lower bound (32.04) by randomly shuffling the appearance matrices and then calculating the CM metric using the shuffled matrices.

Results in Table 1 show that humans outperform the baselines significantly, which indicates that human-written stories match better with input image sequences in terms of characters. We believe that the gap between current SOTA models and the human stories stems from the models' lack of representations for the recurrence of characters in the image sequence.

4 Visually Grounded Story Generation

In this section, we describe the visually grounded story generation model with a new loss function for self-supervised learning of image sequences. In the visual story generation task, given a sequence of images $\mathbf{s} = (img_1, img_2, ..., img_m), \mathbf{s} \in \mathbb{D}^m$, the model needs to generate a story $\mathbf{w}_o = (w_1, w_2, ..., w_{t-1}, w_t), \mathbf{w}_o \in \mathbb{D}^t$.

Encoder-decoder architecture Figure 3 represents our whole architecture, which is based on an encoder-decoder architecture that is also used in most previous work; our architectural changes are represented in the dashed blue box.

The first input to the model is the image sequence $\mathbf{s}$. For the image $img_i \in \mathbf{s}$, we extract global features $f(img_i) \in \mathbb{R}^d$ from the output of the last fully-connected layer of a pre-trained vision backbone model f. We then obtain n_i bounding boxes of object $e_{ij}, j \in [1, n_i]$ predicted by an object detector using the image encoder f as the backbone. We crop each detected object e_{ij} and feed the cropped image region to f to get the local features $f(e_{ij}) \in \mathbb{R}^d$.

The global and local features are first fed to the global encoder f_θ and the local encoder f_ζ corre-

[2] https://vwprompt.github.io/

spondingly in order to project them to the same hidden dimensions as the transformer-based language decoder P_η, to which the hidden representations are fed. To generate tokens auto-regressively, the context $\mathbf{w}$ consisting of previously generated tokens $(w_1, w_2, ..., w_{t-1})$ are fed into a pre-trained text embedding layer Q and then to P_η. Given the input sequence $(\mathbf{s}, \mathbf{w})$, the model represents the probability distribution of the next token w_t as:

$$P(w_t|\mathbf{s}, \mathbf{w}) = P_\eta(f_\theta(f(img_i)), f_\zeta(f(e_{ij})), Q(\mathbf{w}))$$

The major limitation of the encoder-decoder architecture is that there are no representations in both global and local encoders for recurring characters in the input image sequence which we believe are essential to visual coherence in image sequences. The parameters η in the language generator P_η are not sufficient to capture character re-occurrence among the local representations $\mathbf{L_{ij}} = f_\zeta(f(e_{ij}))$. So the visual coherence present in the image sequence is not reflected in textual coherence. We hypothesize that a model that captures and represents the visual coherence better would lead to more coherent stories.

4.1 Visual Coherence Loss

We propose Visual Coherence Loss (VCL, in the blue box of Figure 3) as an auxiliary loss to enable the model to capture the visual coherence of image sequences. Our goal is to measure visual coherence in terms of the recurrence of characters.

Centering theory (Grosz et al., 1995) is one way to explain the relationship between character recurrence and narrative coherence. This theory holds that if a pair of adjacent sentences contain the same discourse entities (here: characters), readers are more likely to find the transition between this sentence pair to be coherent. Character recurrence is very common in visual as well as textual narratives. In addition, images in visual narratives can be considered sentences in textual narratives. Inspired by these, we apply centering theory on measuring coherence in visual narratives.

Character instance similarities. To apply centering theory, we first need to find the same-character recurrences, but we do not have labels for characters at test time. Instead, we can take advantage of the similarities between character instances (as bounding boxes of *person* objects) of the same character across different images. We use the similarities as soft labels to capture different instances of that one character without any manual labels.

To compute similarities between character instances, the first step is to obtain their representations. We identify the objects depicting people automatically and use their local representations $\mathbf{L} = f_\zeta(f(e))$. The representations must be in the latent space of the language decoder P_η. So we pass the local representations through the language decoder to get the contextualized local representations as $\mathbf{E} = P_\eta(\mathbf{L})$. The next step is to compute similarities of character instances across all images. We calculate an instance similarity matrix, consisting of dot products of contextualized local representations, as: $\mathbf{S}' = (s_{ijkl}) = (\mathbf{E}_{ik}^{\mathrm{T}}\mathbf{E}_{jl})$, where $k \in [1, n_i], l \in [1, n_j]$.

Character re-occurrence. Another condition of applying centering theory is to be able to identify where the characters actually recur or not in two adjacent images. We first need to identify occurrences of each character instance in each image. We, therefore, compute a matrix of image-instance similarity. We group elements by image along one dimension indexed by k in $\mathbf{S}'$ and compute the average as: $\mathbf{S} = \text{mean}_{k=1}^{k=n_i}(s_{ijkl}) = (s_{ijl})$ where $i \in [1, m], j \in [1, m], l \in [1, n_j]$. Each element s_{ijl} is the similarity between image i and character instance e_{jl} in image j.

We measure visual coherence in terms of the squared difference between similarities of character instances in the two adjacent images. For an illustration of this idea, consider the following three cases: 1) if the similarity between a character in the present and the previous image is high and the similarity between a character in the present and the next image is low, it implies a sharp transition, the difference is high; 2) if both of them are high, it implies a smooth transition, the difference is low; 3) if both of them are low, it suggests that this character does not appear in either of the two images, the difference is also low. We can thus use the squared difference to measure the level of recurrence of characters across images. Finally, we can compute the average squared difference for all character instances in all pairs of adjacent images as the visual coherence coefficient. We let $\mathbf{S}_{a...b}$ be the rows from a to b indexed by i of $\mathbf{S}$. We can define the coefficient as $\mathcal{L}_\mathbf{C} = \|\mathbf{S}_{2...m} - \mathbf{S}_{1...(m-1)}\|_2$.

Loss function. The lack of local coherence in the visual embeddings is detrimental to the ability of the latent representations in the language to produce coherent stories. We address this by defining a visual coherence loss to encourage the

Model	LF	CM	B-1	B-2	B-3	B-4	M	R-L	C
Seq2Seq		68.03	38.65	20.28	9.78	4.68	31.64	24.24	1.66
Seq2Seq	obj	67.41	40.65	21.35	10.20	4.87	31.69	24.05	1.85
TAPM		67.33	39.85	21.70	10.72	5.19	32.38	25.09	1.48
TAPM	obj	67.99	40.86	22.13	10.83	5.25	32.34	24.91	1.82
Ours									
VCL	obj	70.13	46.83	25.4	12.41	5.84	**33.40**	24.78	3.55
Oracle									
Seq2Seq	oracle	**74.00**	**48.23**	**26.26**	**12.69**	**6.01**	33.23	**24.94**	**4.26**

Table 2: Comparison of all models using different local features (LF) on the test set of VWP using reference-based metrics including BLEU (B), METEOR (M), ROUGE-L (R-L), and CIDEr (C) and our character matching metric (CM). All numbers are an average of three runs with different random seeds.

latent representations to be coherent and maintain the coherence structures in the visual narratives.

We design a ranking loss function to punish stand-alone instances and reward instances that appear consecutively in image visual narratives. To do so, we first construct negative samples by randomly shuffling images in the visual narratives which breaks existing coherent groups of recurrent characters. Then we compute the visual coherence coefficient for negative samples as $\mathcal{L}_C^-$. Lastly, we define the visual coherence loss as a ranking loss: $\mathcal{L}_\mathbf{R} = max\{0, 1 - \mathcal{L}_C^+ + \mathcal{L}_C^-\}$ where $\mathcal{L}_C^+$ is the visual coherence coefficient for positive samples. The final loss function is $\mathcal{L} = \mathcal{L}_\mathbf{L} + \alpha\mathcal{L}_\mathbf{R}$ where $\mathcal{L}_\mathbf{L}$ is the cross-entropy loss for language modeling and α is the weight of ranking loss.

Model training. To better adapt the latent space of model parameters to fit the image sequence, we pre-train our model to optimize the visual coherence loss with image sequences only. We can do so because both terms in the visual coherence loss can be computed with images only. After pre-training, we fine-tune our model with paired image sequences and stories to optimize the final loss.

5 Experiment and Evaluation

In this section, we experiment with the Visual Writing Prompt (VWP; Anonymous, 2023) dataset to demonstrate the effectiveness of our loss function (see Section 3.1). We follow their settings to separate the data into train, validation and test split. We extract global features for all images using the Swin Transformer (Liu et al., 2021), a state-of-the-art supervised vision backbone model. We use the *base* model pre-trained on the ImageNet-21K dataset released on Hugging Face Models. For local features, object features are obtained using a Cascade Mask R-CNN object detector (Cai and Vasconcelos, 2019) with the same Swin Transformer backbone. We crop the bounding boxes of the top 20 objects that the detector predicts for each image. Then we extract the object features (*obj*) similarly as how we extract the global features. We also construct the oracle (*oracle*) features by passing the IDs of the characters to the language decoder. We encode the character IDs via the positions of the feature vectors. For example, character #1 takes the 1st position of the character feature matrix. Because the positional encodings in the language decoder are sensitive to the positions, the character IDs are also encoded in the input.

We use GPT-2 (Radford et al., 2019), a Transformer-based language model pre-trained on large-scale text as the language decoder. We use the small version which is widely used in previous works of story generation. We compare baseline models using the visual coherence loss combined with different character features against the two baseline models defined in Section 3.1.

5.1 Automatic Metrics

To check whether referring expressions of human characters in stories are generated correctly, we apply our character-matching (CM) metric defined in Section 3. To compare with previous baselines, we also evaluate our models with reference-based metrics including unigram (B-1), bigram (B-2), trigram (B-3), and 4-gram (B-4) BLEU scores (B; Papineni et al., 2002), METEOR (M; Banerjee and Lavie, 2005), ROUGE-L (R; Lin, 2004), and CIDEr (C; Vedantam et al., 2015), which were used in the visual storytelling shared task (Mitchell et al., 2018).

Results in Table 2 show that our model trained with visual coherence loss (VCL) substantially outperforms previous baselines like Seq2Seq and

Model	LF	F	M	W	Sum
TAPM	obj	9	26	40	75
VCL (ours)	obj	5	24	31	60
Seq2Seq	oracle	5	19	28	52

Table 3: Number of the three types of character errors in the 50 annotated stories. F, M and W denote Too Few REs, Too Many REs and Wrong REs respectively, as defined in Section 5.2.

TAPM on the CM metric. This supports the hypothesis that the visual coherence loss is an effective means to make the texts better reflect the visual coherence of recurrent characters in the image sequence. We also observe that the model outperforms the baseline on most reference-based metrics except ROUGE-L. The Seq2Seq model with oracle features performs best which again shows the importance of character information.

5.2 Human Evaluation

As we mention in Section 1, we find that most errors are related to lack of local coherence (60% of stories). To get a better understanding of how model fails, we subdivide these errors into three types based on character consistency between images and stories:

Wrong Referring Expressions (REs): Consistent number of characters in the images and stories, but incorrect reference.

Too Few REs: The number of characters in the stories is smaller than the number in the images.

Too Many REs: The number of characters in the stories is more than the number in the images.

We ask three graduate students with a sufficient level of English to annotate these three types of errors. The annotation was preceded by a training phase, during which annotators were instructed to familiarize themselves with the error types and to perform trial annotations. Then, we collected their annotations and gave them feedback to correct for misunderstandings and biases. Formal annotations started after the completion of the training phase, and was blind to condition, i.e., annotators did not know which story was generated by which model. We collected 50 annotated stories for each model.

For each generated story, we counted the occurrence of the three types of character errors and then calculated the mean inter-annotator agreement (IAA) using Pearson's r and Cohen's κ. IAA on this task is moderate $r(48) = 0.45$, $p < 0.01$ and Cohen's kappa coefficient $\kappa = 0.42$.

Feature	Diff↓	Same↑	Δ↑
Swin Transformer			
body	16.41	79	<u>62.59</u>
head ← body	<u>15.41</u>	74.95	59.54
upper body ← body	16.44	78.24	61.8
face	17.85	<u>**79.47**</u>	61.63
body ← face	15.44	77.66	62.22
upper body ← face	16.44	78.29	61.84
head ← face	15.84	77.71	61.87
insightface	**1.07**	67.73	**66.65**
ID	0	100	100

Table 4: Comparison of different vision models for character representations. Diff is the average cosine similarities (%) of different instances of the same character. Same is the average similarities between different characters. Underlined numbers are the best using the same backbone and bold numbers are the best over all.

Results in Table 3 show that compared with the state-of-the-art models, our model makes much fewer character errors. On the other hand, there is still a large gap between our model and one with oracle features, both in Table 2 and Table 3. The global/obj features are not sufficient to represent characters. We thus explore different character features in the next section.

6 Character Representation

Another possible cause of local incoherence is that previous models mostly consider only the global or local visual features without a specific focus on human characters. The only local features that are related to human characters are object features, i.e., the bounding boxes or masks around the *person* objects from the object detector. Object features are effective and widely used in visual story generation. However, these features are either extracted using a general vision backbone model trained on the ImageNet dataset or directly from the output of the last hidden layer of the object detector. These features do not represent human age, gender, face, or body shape. These features can thus not distinguish between different instances of the same human character, which introduces wrong mentions of characters in generated stories.

To improve character representations, we need the following: **1.** distinguishing different characters; **2.** identifying different instances of the same characters. We claim that with these two properties, the visual encoder can represent the characters effectively, and the language decoder can capture

LF	CM	B	M	R-L	C
obj	70.12	**22.66**	32.90	**24.88**	**4.19**
obj, body	70.11	22.16	**32.92**	24.64	3.65
obj, insightface	**71.52**	21.34	32.86	24.36	2.48

Table 5: Results of VCL model with different local features (LF) on the test set of VWP using reference-based metrics including BLEU (B), METEOR (M), ROUGE-L (R-L), and CIDEr (C) and our character matching metric (CM).

these characteristics and generate more coherent and visually-grounded stories.

6.1 Types of Features

To identify characters, the use of facial features seems most promising, as face detectors have been shown to achieve highly accurate results (Deng et al., 2020). To avoid problems due to covered or invisible faces, we also consider features from other body parts. For visual narratives, it is often safe to assume that clothing over other body parts doesn't change between the images of a narrative. Joon Oh et al. (2015) employs the full body features and also inferred upper body features and head features to help person re-identification in photo albums. Body features have been proven to be effective on other similar tasks like character re-identification (Yu et al., 2020), social relation recognition (Sun et al., 2017), and movie description (Rohrbach et al., 2017).

Inferred Features. In our case, we cannot guarantee that our gold labels of human characters cover full bodies, because our annotations are based on the movie shots which often only contain half bodies or heads of the characters.

For each person bounding box, we first pass it to the face detector to get the face bounding box. We propose to use the top boundary of the character bounding box and the other three boundaries of the detected face from face detection to infer a head bounding box. Then we crop the images to get the head features (*head←face*). We follow Joon Oh et al. (2015) to compute body (*body←face*) and upper body (*upper body←face*) features using the inferred head features.

6.2 Analysis of Properties

After obtaining the face and body features, the first question that we would like to investigate is whether these features can distinguish different instances of the same character and also can distinguish different characters. Therefore, we compute the average similarities of different instances of the same character (**Same**) and the average similarities between different characters (**Diff**). We further compute differences between these two similarities (Δ) to show their ability on representing characters.

The results in Table 4 show that general vision backbone models like Swin Transformer are insufficient for differentiating different characters because the Diff scores of Swin-based features are much higher than the insightface features. The most effective method is to use the hidden representations directly extracted from the insightface Face detection system[3].

6.3 Ablation Study

To further compare the best-performing features extracted with Swin Transformer (body) and features based on insightface, we conduct experiments on VWP dataset. We apply different character features one-by-one together with object features to VCL model in visual story generation. We evaluate the models with the same metric as Section 5 except we report the BLEU score (B) as the average of BLEU-1 to BLEU-4 scores.

The results in Table 5 show that the model with *body* features performs on par with the model that only uses *obj* on all metrics. However, insightface features improve the character matching score significantly (t-test, $p < 0.05$), proving face features to be effective.

6.4 Case Studies

To better illustrate the effectiveness of our CM metric, we present a case study in Figure 1. According to the error types defined in Section 5.2, we can see that the Oracle, TAPM and VCL model all make the *Too Few REs* error and additionally the TAPM model makes an *Wrong REs* error, while our proposed VCL model with the insightface feature avoids these problems. Furthermore, comparing the results of METEOR and our proposed CM, we find that METEOR does not reflect the errors in referring expressions: although the VCL model makes one reference error, its METEOR score is higher than the VCL model with the insightface feature, even though it does not make any error in referring expressions. In contrast, our proposed CM can faithfully reflect such errors.

[3] https://insightface.ai/

7 Conclusions

We identified a major important problem of current vision-based story generation models, lack of local coherence (characters are not mentioned correctly). We dissect this problem into two aspects: 1) whether the latent visual representations fed to the language decoder are sensitive to the coherent structure in the visual narratives; 2) whether the visual features can distinguish different characters or identify instances of the same character. We then propose a visual coherence loss to constrain the latent visual representations together with the parameters in the language decoder such that they can represent recurrences of characters. Our model trained with visual coherence loss generates more coherent stories and obtains superior performance on both automatic metrics and human judgments. We also find that using the visual features of characters allows us to produce stories with better identification of recurring characters, which are as a result more locally coherent.

8 Limitations

One limitation is that the importance and visual salience of character instances are not measured directly. We plan to settle these in future work. Another limitation of our work is that we only evaluate our visual coherence loss on a single dataset. Whether the VCL can generalize to other datasets remains unexplored. The reason is that many other datasets are collected in a way to exhibit less visual coherence (lower rates of recurring characters). The VIST dataset (Huang et al., 2016) contains fewer human characters per story than VWP. Also, some of the features we are using for character re-identification may not be suitable to other datasets to the same extent (for instance, if the clothing of characters changes between images).

9 Ethics Statement

The VWP dataset we used is publicly accessible. It is described in a paper (Anonymous, 2023) that is published on TACL under the CC-BY license.

The potential risk of this work is that it can be used to generate harmful content. There could be potentially offensive images in VWP dataset because it is based on movies. Our model might suffer from the risk of learning unwanted correlations between these images and offensive words in the underlying language model GPT-2. Because it has been found that there is a significant amount of unreliable or toxic content in the training data of GPT-2 (Gehman et al., 2020). Although we haven't seen any generated in our human evaluations, our proposed model and code are for research purposes only.

Acknowledgement

This research was funded in part by the German Research Foundation (DFG) as part of SFB 1102 "Information Density and Linguistic Encoding" and a Swedish Research Council (VR) grant (2014-39) for the Centre for Linguistic Theory and Studies in Probability (CLASP). Xudong Hong is supported by International Max Planck Research School for Computer Science (IMPRS-CS) of Max-Planck Institute for Informatics (MPI-INF). We sincerely thank the anonymous reviewers for their insightful comments that helped us to improve this paper. We thank students who help us to evaluate the generated stories: Mattes Alexander Warning, Sijie Wu and Trevor Atkins.

References

Nader Akoury, Shufan Wang, Josh Whiting, Stephen Hood, Nanyun Peng, and Mohit Iyyer. 2020. Storium: A dataset and evaluation platform for machine-in-the-loop story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6470–6484.

Anonymous. 2023. Visual writing prompts: Character-grounded story generation with curated image sequences. *Transactions of the Association for Computational Linguistics*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Tom Braude, Idan Schwartz, Alex Schwing, and Ariel Shamir. 2022. Ordered attention for coherent visual storytelling. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3310–3318.

Zhaowei Cai and Nuno Vasconcelos. 2019. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498.

Khyathi Chandu, Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2019. "my way of telling a story": Persona based grounded story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 11–21, Florence, Italy. Association for Computational Linguistics.

Wei Chen, Xuefeng Liu, and Jianwei Niu. 2022. Sentistory: A multi-layered sentiment-aware generative model for visual storytelling. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):8051–8064.

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2250–2260, New Orleans, Louisiana. Association for Computational Linguistics.

Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. 2020. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ruo-Ping Dong, Khyathi Raghavi Chandu, and Alan W Black. 2019. Induction and Reference of Entities in a Visual Story.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Seraphina Goldfarb-Tarrant, Tuhin Chakrabarty, Ralph Weischedel, and Nanyun Peng. 2020. Content planning for neural story generation with aristotelian rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4319–4338.

Barbara J Grosz, Aravind K Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse.

Xudong Hong, Asad Sayeed, Khushboo Mehra, Vera Demberg, and Bernt Schiele. 2023. Visual writing prompts: Character-grounded story generation with curated image sequences. *Transactions of the Association for Computational Linguistics*, 11.

Xudong Hong, Rakshith Shetty, Asad Sayeed, Khushboo Mehra, Vera Demberg, and Bernt Schiele. 2020. Diverse and relevant visual storytelling with scene graph embeddings. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 420–430, Online. Association for Computational Linguistics.

John E Hopcroft and Richard M Karp. 1973. An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231.

Qiuyuan Huang, Zhe Gan, Asli Celikyilmaz, Dapeng Wu, Jianfeng Wang, and Xiaodong He. 2019. Hierarchically structured reinforcement learning for topically coherent visual story generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8465–8472.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, San Diego, California. Association for Computational Linguistics.

Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. 2015. Person recognition in personal photo collections. In *Proceedings of the IEEE international conference on computer vision*, pages 3862–3870.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 687–692, New Orleans, Louisiana. Association for Computational Linguistics.

Nanxing Li, Bei Liu, Zhizhong Han, Yu Shen Liu, and Jianlong Fu. 2019. Emotion reinforced visual storytelling. *ICMR 2019 - Proceedings of the 2019 ACM International Conference on Multimedia Retrieval*, (May):297–305.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Danyang Liu and Frank Keller. 2023. Detecting and grounding important characters in visual stories. In *37th AAAI Conference on Artificial Intelligence*. AAAI Press.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022.

Margaret Mitchell, Ting-Hao 'Kenneth' Huang, Francis Ferraro, and Ishan Misra, editors. 2018. *Proceedings of the First Workshop on Storytelling*. Association for Computational Linguistics, New Orleans, Louisiana.

Yatri Modi and Natalie Parde. 2019. The steep road to happily ever after: an analysis of current visual storytelling models. In *Proceedings of the Second*

Workshop on Shortcomings in Vision and Language, pages 47–57, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Andrew Piper, Richard Jean So, and David Bamman. 2021. Narrative theory for computational narrative understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 298–311, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mengshi Qi, Jie Qin, Di Huang, Zhiqiang Shen, Yi Yang, and Jiebo Luo. 2021. Latent memory-augmented graph transformer for visual storytelling. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4892–4901.

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

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Anna Rohrbach, Marcus Rohrbach, Siyu Tang, Seong Joon Oh, and Bernt Schiele. 2017. Generating descriptions with grounded and co-referenced people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4979–4989.

Qianru Sun, Bernt Schiele, and Mario Fritz. 2017. A domain based approach to social relation recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3481–3490.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4566–4575.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Ruize Wang, Zhongyu Wei, Piji Li, Qi Zhang, and Xuanjing Huang. 2020. Storytelling from an image stream using scene graphs. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9185–9192. AAAI Press.

Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 899–909, Melbourne, Australia. Association for Computational Linguistics.

Peng Xu, Mostofa Patwary, Mohammad Shoeybi, Raul Puri, Pascale Fung, Anima Anandkumar, and Bryan Catanzaro. 2020. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2831–2845, Online. Association for Computational Linguistics.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Licheng Yu, Mohit Bansal, and Tamara Berg. 2017. Hierarchically-attentive RNN for album summarization and storytelling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 966–971, Copenhagen, Denmark. Association for Computational Linguistics.

Youngjae Yu, Jiwan Chung, Heeseung Yun, Jongseok Kim, and Gunhee Kim. 2021. Transitional adaptation of pretrained models for visual storytelling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12658–12668.

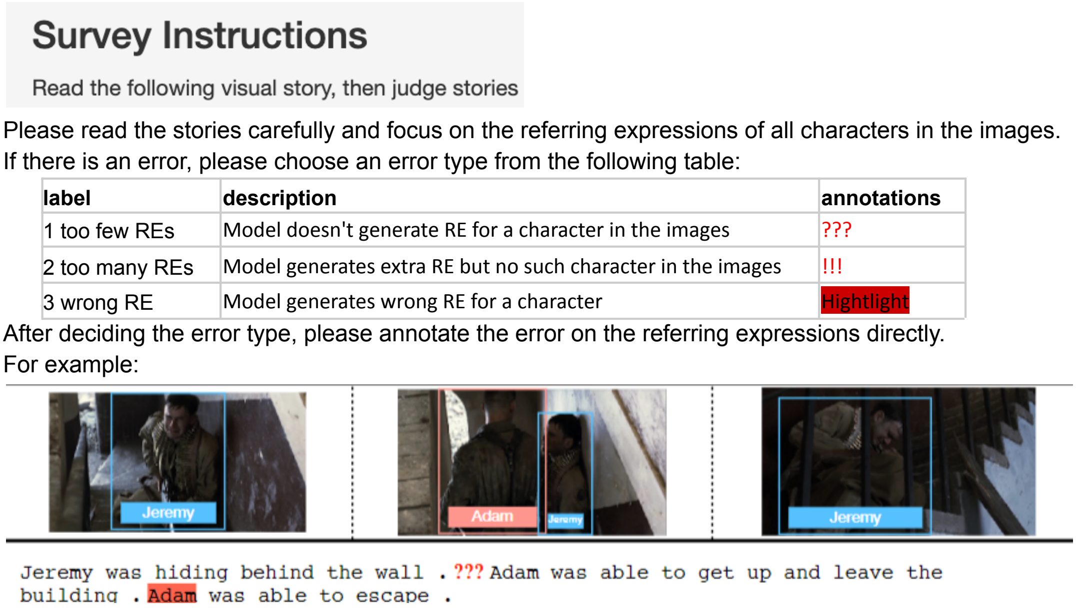
Youngjae Yu, Jongseok Kim, Heeseung Yun, Jiwan Chung, and Gunhee Kim. 2020. Character grounding and re-identification in story of videos and text descriptions. In *European Conference on Computer Vision*, pages 543–559. Springer.

A Settings of Human Evaluation

For the self-assessment in Section 1, we first adopt the list of common errors from Modi and Parde (2019). Then we add additional errors that we identified. We also merge errors that are too similar for simplicity. Finally, we define nine error types which are described in Table 6. After that, one

Error Type	Description
Too generic	Lack of semantic variations. E.g. too many "good/great time", "happy".There are biases in the dataset.
Wrong event	Incorrect recognition of the event: due to the context
Wrong object	Incorrect recognition of the event participant
Grammatical error	Syntactic error: using noun phrases (NPs), missing function words, and others...
Repetition	Duplicated: a later utterance has almost the same meaning with an earlier utterance
Semantically inconsistent	Semantically inconsistent: 1) Violation of selectional preferences 2) Missing entity
Wrong character	Discourse coherence: incorrect use of anaphora
Lack of script knowledge	The connections between events contradict commonsense knowledge
Lack of narrativity	The text doesn't not contain a plot

Table 6: Error types of human evaluation and results of a SOTA model.



Survey Instructions

Read the following visual story, then judge stories

Please read the stories carefully and focus on the referring expressions of all characters in the images.
If there is an error, please choose an error type from the following table:

label	description	annotations
1 too few REs	Model doesn't generate RE for a character in the images	???
2 too many REs	Model generates extra RE but no such character in the images	!!!
3 wrong RE	Model generates wrong RE for a character	Hightlight

After deciding the error type, please annotate the error on the referring expressions directly.
For example:

Jeremy was hiding behind the wall . ??? Adam was able to get up and leave the building . Adam was able to escape .

Figure 4: Survey instructions.

author annotated 50 stories generated by the SOTA model TAPM.

For the human evaluation in Section 5.2, the annotation guideline is in Figure 4. We process all the data we collect to delete unique identities of annotators. One annotator are paid by the funding source under a student assistant contract. The other two are student volunteers. One annotator is a native speaker of English and the other two are graduate school students of Saarland University who speak proficient English.

B Settings of Experiments

For the analysis in Section 3.1, We use this implementation of a coreference resolution system[4] from AllenNLP. We use this version[5] of Scipy.

For the experiments in Section 5.2, We build our system base on the code[6] by Yu et al. (2021). We follow their settings for hyper-parameters as initial setting. Then we tune our hyper-parameters based on evaluation results on validation split. We use the Huggingface transformer for Transformer-based models: Swin Transformers[7] and GPT-2[8]. We use the *base* versions of both Swin Transformers and GPT-2 due to our limited computing resource. E.g. our maximum batch size is 6 on a GPU with 32GB memory. The total computation time is about 6

[4] https://demo.allennlp.org/coreference-resolution

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csgraph.maximum_bipartite_matching.html

[6] https://github.com/JiwanChung/tapm/

[7] https://huggingface.co/microsoft/swin-base-patch4-window12-384-in22k

[8] https://huggingface.co/gpt2

hours for 20 epochs of training on one of our Nvidia V100 32GB GPU. Our code is in Supplemental, which is unlicensed and for review only. All hyper parameters are in *config.py*. We will publish our code on Github after the review process of this submission.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Section 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

We use scientific artifacts in Sections 3, 4, 5, and 6. The scientific artifacts we create are listed at the end of Section 1 and described in Sections 3 and 4.

☑ B1. Did you cite the creators of artifacts you used?
All creators of artifacts that we use are cited properly across the paper. All cited artifacts are listed in References section.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 9

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 9

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Appendix

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Appendix

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3 and 5

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sections 3, 5, and Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3, 5.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Sections 3, 5, and Appendix

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Section 5.2

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 5.2 and Appendix

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Appendix

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Section 5.2

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
There is no potential ethic issues in the data collection process.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix