Supplementary Material for StoryDALL-E: Adapting Pretrained Text-to-Image Transformers for Story Continuation

Adyasha Maharana, Darryl Hannan, and Mohit Bansal

UNC Chapel Hill, NC 27514, USA
{adyasha,dhannan,mbansal}@cs.unc.edu

Overview

The supplementary material in this PDF is organized as follows:

- **Section 1**: Background information on the story visualization task and DALL-E Transformer models.
- **Section 2**: Details of the StoryDALL-E and StoryGANc models.
- **Section 3**: Datasets and their construction.
- **Section 4**: Implementation details, including training hyperparameters, evaluation metrics and pretrained checkpoints.
- **Section 5**: Results on validation sets of story continuation datasets.
- **Section 6**: Analysis experiments including the comparison of story visualization and story continuation tasks, correlation scores between source and generation, retrieval-based text-to-image synthesis and analysis of the semantic content of the DiDeMoSV datasets.

1 Background

In this section, we give a brief introduction to the original story visualization task and auto-regressive transformers for text-to-image synthesis.

1.1 Story Visualization

Given a sequence of sentences $S = [s_1, s_2, ..., s_T]$ forming a narrative, story visualization is the task of generating a corresponding sequence of images $\hat{X} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_T]$, following [8]. The sentences form a coherent story with recurring plot and characters. The generative model for this task has two main modules: story encoder and image generator. The sentence encoder $E_{caption}(.)$ takes word embeddings $\{w_{ik}\}$ for sentence s_k at each timestep k and generates contextualized embeddings $\{c_{ik}\}$. These embeddings are then used to generate the corresponding images. The terms *caption* and *sentence* are used interchangeably throughout the paper.

1.2 Pretrained Text-to-Image Synthesis Models (DALL-E)

The DALL-E model introduced in [15] is a text-to-image synthesis pipeline which comprises of a discrete variational autoencoder (dVAE) in the first stage and an autoregressive transformer in the second stage:

Stage 1. The Vector Quantized Variational Autoencoder (VQVAE) [13] consists of an encoder that learns to map high dimensional input data (x) to a discretized latent space, and a decoder that reconstructs x from the quantized encodings x^q. The model is trained using the reconstruction loss and commitment loss [20]. In DALL-E, the VQVAE is trained to transform RGB image into a small 2D grid of image tokens, where each token can assume a discrete value from a codebook of predefined length.

Stage 2. The VQVAE encoder from Stage 1 is used to infer the grid of discretized image tokens which is flattened and concatenated with the input text tokens, and an autoregressive transformer is used to model the joint distribution over the text and image tokens. For a given text input s and target image x, these models learn the distribution of image tokens $p(x)$ as,

$$p(x) = \prod_{i=1}^{d} p(x_i | x_{i<i}; s) | x < i) \tag{1}$$

The models are composed of stacked multi-head self-attention layers with causal masking and are optimized via maximum likelihood. Each self-attention block is followed by a MLP feedforward layer, as per the standard design of transformers. The prediction of image tokens at each time step is influence by the text tokens and previously predicted image tokens via the self-attention layer.

Using this framework, DALL-E obtains impressive, state of the art results on a variety of text-to-image tasks by leveraging large scale pre-training on multimodal datasets.

2 Additional Method Details

In this section, we provide additional details about the StoryDALL-E and StoryGANc models.

2.1 StoryDALL-E

Retro Cross-Attention Layer Density. We experiment with different densities of cross-attention layers in our implementation of StoryDALL-E. In the most dense variation, we introduce the retro layer in every self-attention block of minDALL-E, effectively increasing the number of parameters in the model by nearly 60%. We vary the density of the retro layer for one in every 1-5 self-attention block(s), and run experiments for each of these variations. Our best model has a density of one retro layer in every third self-attention block.

Objective. Following the original DALL-E implementation [15], the StoryDALL-E model is trained on a combination of text loss and image loss. The losses are cross-entropy losses for the respective modalities, and the combined objective is,

$$\mathcal{L} = -\sum_{i=1}^{N_{text}} t_i log(p(t_i) - \sum_{i=1}^{N_{img}} m_i log(p(m_i)$$

where N_{text} and N_{img} are the caption lengths and image sequence lengths, set to 64 and 256 in our model respectively.

2.2 StoryGANc

StoryGANc follows the general framework of the StoryGAN model [8] i.e., it is composed of a recurrent text encoder, an image generation module, and two discriminators - image and story discriminator. We modify this framework to accept the source frame as input for the story continuation task, and use it for improving the generation of target frames. Our StoryGANc model is implemented as follows:

Pre-trained Language Model Encoder. In the current state-of-the-art story visualization models [12], recurrent transformer-based text encoders like MART [7] and MARTT [11] are learnt from scratch for encoding the captions. However, while the memory module contains information about prior captions, there is no way for the current caption to directly attend to words in prior or subsequent captions. This is crucial in a story where causality plays such a large role, e.g., which characters need to appear in the scene, even if they don't appear in the current caption, has there been any modifications to the background that need to appear in the current scene, etc. Furthermore, general world knowledge is crucial for successfully generating unseen stories in our datasets, which is possible with pretrained knowledge. Therefore, we propose using a pretrained language model (such as RoBERTa [10] or CLIP text encoder [14]) as the caption encoder. These models are pretrained on large unimodal or multimodal datasets of language; their latent knowledge of the world is of great utility for understanding the semantic concepts present in input captions. For the RoBERTa encoder [10], to ensure that the model has access to all captions, we append the captions together and feed all of them into each timestep. We use a special token to denote which caption is currently being generated. The representation from the first token h_0 is used as the caption representation. For the CLIP encoder [14], we add an additional self-attention block that takes the caption representation for each timestep and produces the contextualized representations that have been computed by attending to all other timesteps.

Contextual Attention. We then combine the story representation with the image embeddings of the first frame of the image sequence using contextual attention. First, we reshape the story representation as a 2D matrix and extract 3×3 patches $\{t_{x,y}\}$ as convolutional filters. Then, we match them against potential

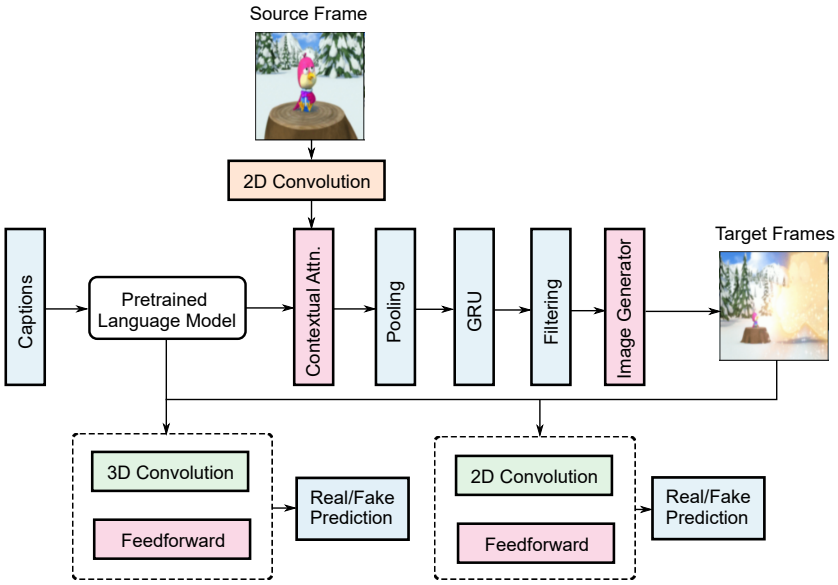

Fig. 1. Illustration of our StoryGANc architecture. The captions are first encoded using a pretrained language model to produce contextualized representations. These representations are sent to a contextual attention module along with the source frame, and the resulting representation is sent to the image generator. The generated frames are sent to a story and image discriminator, and the corresponding cross-entropy losses for detection real/fake images are used to train the StoryGANc model.

patches from the source frame $\{s_{x',y'}\}$ by measuring the normalized inner product as,

$$p_{x,y,x',y'} = \langle \frac{s_{x,y}}{||s_{x,y}||}, \frac{t_{x',y'}}{||t_{x',y'}||} \rangle \tag{2}$$

where $p_{x,y,x,y'}$ represents the similarity between the patch centered in target frame (x, y) and source frame (x', y'). We compute the similarity score for all dimensions along (x', y') for the patch in target frame (x, y) and find the best match from the softmax-scaled similarity scores. [21] implement this efficiently using convolution and channel-wise softmax; we use their implementation in our StoryGANc model. The extracted patches are used as deconvolutional filters and added to the target frame s. The resulting representation is fed through a generator module which processes each caption and produces an image. We use the generator module outlined in [8].

Discriminators. Finally, the loss is computed for the generated image sequence. There are 3 different components that provide the loss for the model. The first is a story discriminator, which takes all of the generated images and uses 3D convolution to create a single representation and then makes a prediction as to whether the generated story is real or fake. Additionally, there is an image

discriminator, which performs the same function but only focuses on individual images. Finally, the model is trained end-to-end using the objective function:

$$\min_{\theta_G} \max_{\theta_I, \theta_S} \mathcal{L}_{KL} + \mathcal{L}_{img} + \mathcal{L}_{story}$$

where θ_G, θ_I and θ_S denote the parameters of the text encoder + generator, and image and story discriminator respectively. $\mathcal{L}_{img}$ and $\mathcal{L}_{story}$ are cross-entropy losses for classifying ground truth and synthetic images into real and fake categories respectively. $\mathcal{L}_{KL}$ is the Kullback-Leibler (KL) divergence between the learned distribution h_0 and the standard Gaussian ditribution, to enforce smoothness over the conditional manifold in latent semantic space [8]. During inference, the trained weights θ_G are used to generate a visual story for a given input of captions.

3 Dataset Construction

We propose the new dataset DiDeMoSV, which is derived from the Didemo dataset [6]. Below, we present details about collection and cleaning of the dataset.

3.1 Dataset Construction

Prior work in story visualization has repurposed datasets from other tasks. We follow this trend and repurpose video captioning datasets in our work. Story visualization and video captioning share many components. In video captioning, an agent must produce a caption, or series of captions, that describe the content of a video. Story visualization can be thought of as video captioning in reverse, where frames are generated based upon the captions. However, simply reversing the direction of the task is not sufficient in this case because the other difference between the two tasks is that story visualization has one frame per caption, whereas videos have many frames; a single caption is typically paired with a video time stamp, denoting which section of the video the caption aligns with. Therefore, to convert video captioning into story visualization, an appropriate method is needed to select which single frame should be used to represent the content of the caption.

We employ the self critical image captioning model [17] for intelligently selecting the frame most aligned with the caption. Each of the clips that correspond to a caption is multiple seconds long. Not all of the frames will be equally aligned with the caption. Characters might be moving leaving blur effects, the scene might change a bit early or late in the clip, or there might be superfluous actions that occur. To initially shrink the number of frames that we must consider, we first sample frames at fixed intervals throughout the video. In the case of DiDeMoSV, we sample 10 frames. Each of the frames is then fed through the self critical image model and is ranked according to the sum of the log likelihood for each word in the caption being generated. We then use the top ranked frame as the image for the given caption. The resulting image-caption sequence after

this step is on average 4 frames long for DiDeMoSV. To maximize the amount of data that we have and make the task feasible, we split these image-caption sequences into a sequence of 3 frames. We use a sliding window approach to create these sequences, allowing for overlap between sequences. However, we also ensure that the train, val, and test splits contain separate videos. We then proceed with our image pre-processing steps.

The main pre-processing step that we explore is to convert the real-world images into cartoon images, to emphasize focus on the main characters of the image rather than the trivial details of the background. Rather than models focusing on making images realistic, we want them to focus on accurately representing the stories themselves in visual form. To cartoonize the images we use CartoonGAN [4]. Each of the extracted frames is fed through this network and the resulting output is used in the final dataset.

4 Experimental Details

Pretrained Weights. While the VAE checkpoints for the original DALL-E model have been released, the transformer weights have not. We explored training the transformer component from scratch on our data, but found that it did not perform well. Therefore, we explored other publicly available efforts to reproduce DALL-E and settled on a popular open-source version minDALL-E which is composed of 1.3 billion parameters and trained on 14 million text-image pairs from the CC3M [19] and CC12M [3] datasets.[1] minDALL-E uses the pretrained VQGAN-VAE [5] for discretizing image inputs. We adapt the pretrained model minDALL-E to StoryDALL-E and then prompt-tune/fine-tune the retro-fitted model on our target datasets.

We experiment with pretrained CLIP [14] (38M parameters) and distilBERT [18] (110M parameters) text encoders for the LM-StoryGAN models. The CLIP image encoder is used to extract image embeddings for the source frame in thes tory continuation task. The universal sentence transformer [2] is used to extract sentence embeddings for captions, that are sent as input to the global story encoder in StoryDALL-E.

Training Details. We conduct experiments in the story continuation setting, i.e., the models receive the first frame as input condition. The StoryDALL-E models are trained for 5 epochs with learning rates of 1e-04 (AdamW, Cosine Scheduler) and 5e-04 (AdamW, Linear Decay Scheduler) for fine-tuning and prompt-tuning setups respectively. We use a cosine schedule with warmup from 0 in the first 750 training steps. The minimum learning rate is 0.1 times the maximum learning rate. Checkpoints are saved at the end of every epoch. In full-model finetuning settings, the pretrained weights are finetuned with a smaller learning rate of 1e-05. The LMStoryGAN models are trained for 120 epochs with learning rates 1e-04 and 1e-05 for the generator and discriminators respectively. Checkpoints are saved every 10 epochs. These models are trained on single A6000 GPUs.

[1] https://github.com/kakaobrain/minDALL-E

Table 1. Results on the validation sets of PororoSV, FlintstonesSV and DiDeMoSV (DSV) datasets from various models. Scores are based on FID (lower is better), character classification F1 and frame accuracy (F-Acc.; higher is better) evaluations.

Model	PororoSV			FlintstonesSV			DSV
	FID ↓	Char-F1↑	F-Acc↑	FID ↓	Char-F1↑	F-Acc↑	FID↓
StoryGANc (BERT)	63.94	54.02	24.53	87.65	71.98	55.68	93.21
StoryGANc (CLIP)	65.13	**54.83**	**25.29**	87.02	72.30	**59.35**	93.26
StoryDALL-E (prompt)	45.68	31.91	22.14	67.05	54.17	26.23	72.61
StoryDALL-E (finetuning)	**21.64**	40.28	20.94	**28.37**	**74.28**	52.35	**41.58**

Evaluation Metrics. We consider 3 automatic evaluation techniques. The first is FID score, which calculates the difference between the ground truth and generated images by computing the distance between two feature vectors. We follow prior work and use Inception-v3 as our image encoding model.

Following [8] and [12], we also compute the character classification scores for the Pororo and Flintstones datasets, which are adapted from video QA datasets with recurring characters. We use the Inception-v3 models trained for character classification on these respective datasets for computing the F1 Score and frame accuracy (exact match). Since the DiDeMoSV dataset does not have recurring characters, we do not evaluate performance of our models on these datasets using character classification.

5 Additional Results

In this section, we present the results on validation sets of the three story continuation datasets discussed in Table 2 in main text.

Validation Set Results. We present results on the validation set of the three story continuation datasets discussed in main text i.e. PororoSV, FlintstonesSV and DiDeMoSV, in Table 1. The fully-finetuned StoryDALL-E model performs the best across all datasets in terms of FID score. The gains are seen in FID, due the high visual quality of the images generated by StoryDALL-E. However, the character classification anf frame accuracy scores for the StoryDALL-E are close to those of StoryGANc for the FlintstonesSV dataset and relatively lower for the PororoSV dataset, in spite of being of better visual quality (as per manual analysis). This might be attributed to that fact that GAN-based models tend to generate some finer details of a character while sacrificing shape and form, which is recognized by character classification models as a faithful reconstruction. On the other hand, StoryDALL-E models focus on shape and form and tend to blur other defining characteristics, which are appealing to human eyes, but fail to be recognized by the classification model.

Due to the higher resolution images generated by VQGAN-VAE [5], the visual quality of images produced by StoryDALL-E is highly preferred over predictions from the StoryGANc models. Similarly, the latent pretrained knowledge

of DALL-E promotes generation of images that align well with the input captions, and results in higher wins for the StoryDALL-E model. The %wins and %loss are nearly uniform for the attribute *consistency* in this larger experiment, for the PororoSV and DiDeMoSV datasets. Predictions from the StoryDALL-E model are found to be more consistent than those of StoryGANc for the FlintstonesSV dataset. See predictions from StoryDALL-E for the PororoSV, FlintstonesSV and DiDeMoSV datasets in Figures 5, 6 and 7 respectively.

6 Additional Analysis

In this section, we examine various aspects of the story continuation task, models and datasets. First, we demonstrate the advantages of the story continuation task over the story visualization task. Next, we calculate correlations between the source images and generated images from StoryDALL-E, with and without condition, to demonstrate the utility of cross-attention layers. Then, we examine the effect of the retro-fitting approach in a text-to-image synthesis task. Finally, we discuss the semantic content of our proposed DiDeMoSV dataset.

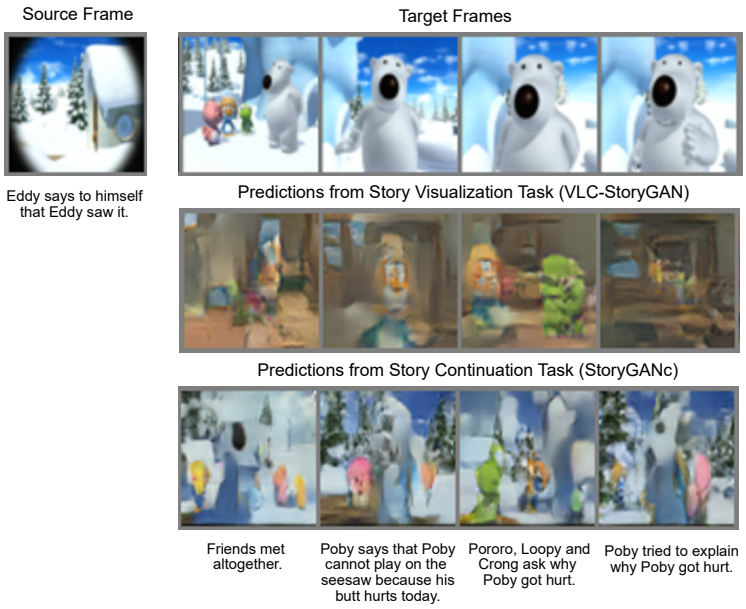

Fig. 2. Comparison of predictions from state-of-the-art story visualization model VLC-StoryGAN (middle) and our story continuation model StoryGANc (bottom) for a sample from the PororoSV dataset (top).

6.1 Story Visualization vs. Story Continuation

In Figure 2, we present a comparison of predictions from the state-of-the-art story visualization model VLCStoryGAN [11] and our story continuation model StoryGANc for a sample from the test set of the PororoSV dataset. Story Visualization relies only on the input captions to generate the images from scratch. However, as discussed in Section 3.1 in the main text, the captions in story visualization datasets are short and do not contain information about the setting and background elements. As a result, the predictions from story visualization models rely on data seen in the training set to infer arbitrary visual elements. In Figure 2, the story takes place in a snowy field with trees (top), but the prediction from VLCStoryGAN (middle) depicts the story as taking place indoors. When the first frame is given as additional input to our model StoryGANc in the story continuation task, the models borrows the snowy fields from the source frame and creates the story within that setting (bottom). Hence, story continuation is a more realistic and practical version of story visualization that can enable significant progress in research and faster transfer of technology from research to real-world use cases. Our experiments and datasets demonstrate the utility of this task.

6.2 Correlation between Source and Generated Images

We also measure the cosine similarity between the source frames and the generated frames from StoryDALL-E, with and without the retro-fitted cross-attention layer for conditioning on a source image, as a representation of the correlation between the two sets of images. We encode the images using the CLIP image encoder ViT-B/16 and report the mean and standard deviation of cosine similarity values for each dataset (see Table 2). We see upto 0.3 points increase in correlation between source image and generated image for all three datasets with the use of the conditioning mechanism.

Table 2. Mean and standard deviation of correlation between source image and generated images from StoryDALL-E without and with conditioning on the source image.

Dataset	without condition	with condition
PororoSV	0.23 ± 0.04	0.26 ± 0.04
FlintstonesSV	0.38 ± 0.05	0.41 ± 0.03
DiDeMoSV	0.16 ± 0.04	0.19 ± 0.01

6.3 Retrieval-based Text-to-Image Synthesis

[1] show that retrieving nearest-neighbor sentences during prediction of the next token in a sentence can improve generation from smaller GPT models, and

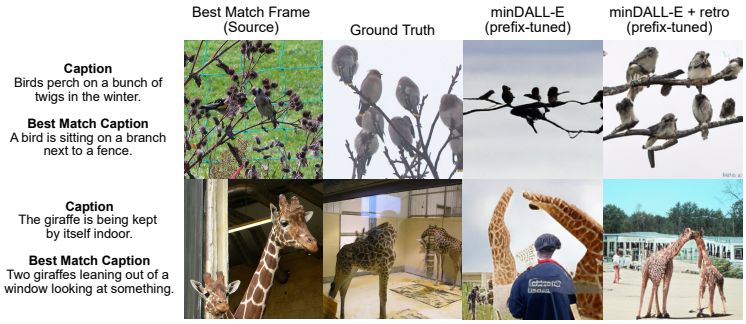

Fig. 3. Sample predictions for the MS-COCO dataset using prompt-tuned minDALL-E models with and without retro-fitting. The image corresponding to the nearest neighbor caption is used as source frame for the retro-fitted model.

bring their performance close to the 10x larger GPT3 models. We use this mechanism to condition the images on a source frame in the story continuation task, however, it can also be used to copy from the nearest neighbor images for the text-to-image synthesis task. Hence, we perform experiments to test this hypothesis. We prompt-tune minDALL-E for text-to-image synthesis on the MSCOCO dataset [9] and compare it to a similar model that is additionally retro-fitted with a cross-attention layer.[2] First, we find the nearest neighbor caption in MSCOCO training set for each caption in the validation set using CLIP [14] embeddings. Next, we use the corresponding image of the nearest neighbor caption and send the image embeddings from VQGAN-VAE as input to the cross-attention layer in retro-fitted minDALL-E. We compute FID scores on the predictions of both models; the minDALL-E model with retro-fitted layers achieves 149.29 score on the validation set of MSCOCO, while the model without retro-fitted layer achieves 155.34 FID score on the same (lower is better). See Figure 3 for comparison of a few sample predictions from both models, along with their ground truths. With the retrieval-based layer, the minDALL-E model is able to recreate semantic concepts like bird (top) and giraffe (bottom) more accurately. This demonstrates the utility of nearest neighbor retrieval and our method of integrating it into pretrained models for text-to-image synthesis.

6.4 Semantic Analysis of the DiDeMoSV dataset.

Figure 4 contains counts for (A) noun chunks, (B) verbs and (C) object classes in DiDeMoSV. As discussed in Section 3, DiDeMoSV is collected from Flickr and the most common nouns indeed reflect this. Most of the captions are descriptive in that they describe the contents of the scene, the location of the objects/people in the scene, and the actions that are taking place in the scene. In DiDeMoSV, the focus is on the breadth of information that must be considered in the form of actions, objects, and settings.

[2] https://github.com/kakaobrain/minDALL-E

The graph for the frequency of verbs across the captions in the DiDeMoSV dataset (see (B) in Figure 4) illustrates the complexity of the actions that are being undertaken by agents in the story. It can be seen that most of the actions are simplistic and related to movement, such as “walks", “comes", “starts", “turns", “goes", etc. A lot of the verbs are also centered around vision, such as “see", “seen", and “looks". While these words corroborate our prior insights reflecting the relative simplicity of the stories in DiDeMoSV, they also are crucial for understanding simplistic event chains. An understanding of these simple verbs and the way that they affect the story goes a long way towards facilitating story continuation, especially in the many settings of DiDeMoSV.

Part (C) in Figure 4 contains a breakdown of the objects that appear in the DiDeMoSV images. To generate these graphs, we use Yolov3 [16] to process each of the images in the respective datasets. The ’person’ class is the dominant class in both datasets. This intuitively makes sense due to the initial data sources from which the respective video captioning datasets were constructed. Additionally, it matches the pattern that is observed in the caption noun analysis, where the nouns in both datasets are most frequently referring to people. However, we can also see that there are limitations of the Yolov3 model. There are frequently occurring nouns, such as ‘camera’ in DiDeMoSV that are not able to appear in our image analysis because these do not have corresponding classes in the model. We use the default confidence threshold of 0.25 in the Yolo model, which generates predictions for only 76% of DiDeMoSV images.

Our analysis demonstrates the diversity of the DiDeMoSV dataset, and showcases it as a challenging benchmark for the story continuation task, in addition to PororoSV and FlintstonesSV.

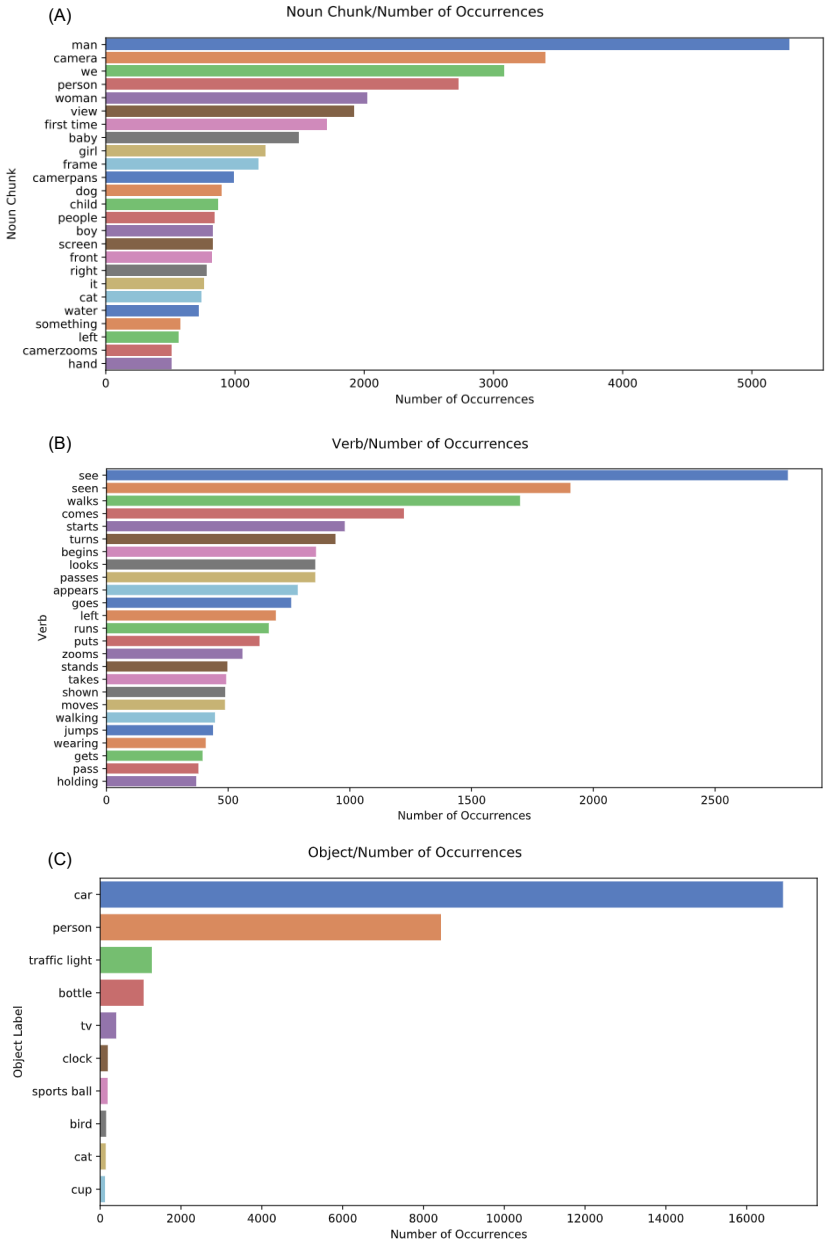

Fig. 4. Plots for frequency of (A) noun chunks and (B) verbs in the captions and (C) objects in the frames of the DiDeMoSV dataset.

Fig. 5. Generated samples from StoryDALL-E for the PororoSV dataset.

Fig. 6. Generated samples from StoryDALL-E for the FlintstonesSV dataset.

Fig. 7. Generated samples from StoryDALL-E for the DiDeMoSV dataset.

References

1. Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., Driessche, G.v.d., Lespiau, J.B., Damoc, B., Clark, A., et al.: Improving language models by retrieving from trillions of tokens. arXiv preprint arXiv:2112.04426 (2021) 9
2. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018) 6
3. Changpinyo, S., Sharma, P., Ding, N., Soricut, R.: Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3558–3568 (2021) 6
4. Chen, Y., Lai, Y.K., Liu, Y.J.: Cartoongan: Generative adversarial networks for photo cartoonization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9465–9474 (2018) 6
5. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12873–12883 (2021) 6, 7
6. Hendricks, L.A., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2017) 5
7. Lei, J., Wang, L., Shen, Y., Yu, D., Berg, T., Bansal, M.: Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2603–2614 (2020) 3
8. Li, Y., Gan, Z., Shen, Y., Liu, J., Cheng, Y., Wu, Y., Carin, L., Carlson, D., Gao, J.: Storygan: A sequential conditional gan for story visualization. In: Proceedings of the IEEE Conference on CVPR. pp. 6329–6338 (2019) 1, 3, 4, 5, 7
9. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 10
10. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019) 3
11. Maharana, A., Bansal, M.: Integrating visuospatial, linguistic, and commonsense structure into story visualization. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 6772–6786 (2021) 3, 9
12. Maharana, A., Hannan, D., Bansal, M.: Improving generation and evaluation of visual stories via semantic consistency. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 2427–2442 (2021) 3, 7
13. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelcnn decoders. Advances in neural information processing systems **29** (2016) 2
14. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning. pp. 8748–8763. PMLR (2021) 3, 6, 10

15. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. In: International Conference on Machine Learning. pp. 8821–8831. PMLR (2021) 2, 3
16. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018) 11
17. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7008–7024 (2017) 5
18. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In: 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (NeurIPS) (2019) 6
19. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018) 6
20. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017) 2
21. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018) 4