Reading Between the Lines: Exploring Infilling in Visual Narratives

Khyathi Raghavi Chandu **Ruo-Ping Dong** **Alan W Black**

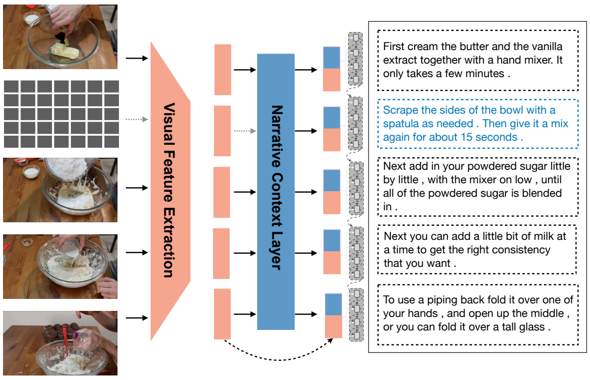
Language Technologies Institute, Carnegie Mellon University
{kchandu, awb}@cs.cmu.edu, ruopingd@alumni.cmu.edu

Abstract

Generating long form narratives such as stories and procedures from multiple modalities has been a long standing dream for artificial intelligence. In this regard, there is often crucial subtext that is derived from the surrounding contexts. The general seq2seq training methods render the models shorthanded while attempting to bridge the gap between these neighbouring contexts. In this paper, we tackle this problem by using *infilling* techniques involving prediction of missing steps in a narrative while generating textual descriptions from a sequence of images. We also present a new large scale *visual procedure telling* (ViPT) dataset with a total of 46,200 procedures and around 340k pairwise images and textual descriptions that is rich in such contextual dependencies. Generating steps using infilling technique demonstrates the effectiveness in visual procedures with more coherent texts. We conclusively show a METEOR score of 27.51 on procedures which is higher than the state-of-the-art on visual storytelling. We also demonstrate the effects of interposing new text with missing images during inference. The code and the dataset will be publicly available at https://visual-narratives.github.io/Visual-Narratives.

1 Introduction

Humans process information from their surrounding contexts from multiple modalities. These situated contexts are often derived from a modality (source) and expressed in another modality (target). Recent advances have seen a surge of interest in vision and language as source and target modalities respectively. One such widely studied task is image captioning (Hossain et al., 2019; Liu et al., 2019) which provides a textual description T given an image I. In contrast, visual storytelling (Huang et al., 2016) is the task of generating a sequence of textual descriptions ($\{T_1, T_2, ..., T_n\}$) from a sequence of images ($\{I_1, I_2, ..., I_n\}$).



First cream the butter and the vanilla extract together with a hand mixer. It only takes a few minutes .

Scrape the sides of the bowl with a spatula as needed . Then give it a mix again for about 15 seconds .

Next add in your powdered sugar little by little , with the mixer on low , until all of the powdered sugar is blended in .

Next you can add a little bit of milk at a time to get the right consistency that you want .

To use a piping back fold it over one of your hands , and open up the middle , or you can fold it over a tall glass .

Figure 1: Overview of infilling in visual procedures. Image in the second step is masked while the model generates the corresponding textual description from surrounding context.

A fundamental incongruity between how humans process information from multiple modalities and how we teach machines to do the same is that, humans are capable of bridging the information gap from surrounding contexts. Traditionally, the problem of missing context in long text generation is addressed using additional input such as entities, actions, etc., (Fan et al., 2019; Dong et al., 2019). In contrast, in the spirit of simplicity, we propose infilling techniques to interpolate the gap between surrounding contexts from a stream of images. We focus on two kinds of visual narratives namely, stories and procedures. We curated a large scale *ViPT* dataset with pairwise image and text descriptions comprising of 46k procedures and 340k images. The percentage of unique words in each step in comparison to the rest of the recipe is about 60% for ViST and 39% for ViPT. This implies that overlapping contexts are predominant in procedures than stories datasets. This is usually because stories are more creative and diverse while procedures are in-domain. For both these reasons, we hypothe-

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1220–1229,
November 16–20, 2020. ©2020 Association for Computational Linguistics

Dataset	**ViST**	**Visual Procedure Telling (ViPT)**									
Categories	**stories**	**recipes**	**crafts**	**outdoors**	**lifestyle**	**technology**	**styling**	**fitness**	**hobbies**	**pets**	**misc**
#narratives	50,136	34,138	660	1,831	1,824	1,660	1,585	911	1,701	858	1,032
#images or steps	209,651	203,519	8,658	20,526	20,959	19,221	18,112	9,935	19,145	9,599	11,853
avg #steps	5.00	5.96	13.12	11.21	11.49	11.57	11.42	10.90	11.25	11.18	11.48
avg #words/step	11.35	79.19	47.99	35.52	32.58	27.90	17.31	17.54	17.54	17.24	57.45

Table 1: Details of the ViST and *Visual Procedure Telling* Dataset broken down into 10 categories

size that infilling technique is more effective in scenarios where it can leverage the vast context from the surrounding information to filling the missing pieces. To this end, we present our infilling based model to perform visual narrative generation and compare its effects on visual stories and procedures. The overview of the infilling bassed training procedure is presented in Figure 1. We conclusively observe that it is more effective in procedural texts with stronger contextual dependencies. Interpolating contexts to generate narrative descriptions has potential applications in fields such as digital education (Hollingshead, 2018), social media content (Gella et al., 2018), augmented reality (Dudley et al., 2018), video games (Kurihara et al., 2019; Ammanabrolu et al., 2019), etc,.

The main contributions of this paper are:

• We present a Visual Procedure Telling (ViPT) dataset similar to the Visual Storytelling (ViST) dataset with 46k procedures on various domains.

• We demonstrate the efficacy of our visual infilling technique on narratives that have stronger contextual dependencies on the rest of the sentences.

2 Related Work

Multimodal Language: Language generation from visual modality has seen a steep rise in interest with the introduction of several large scale tasks such as image captioning (Hossain et al., 2019), visual question answering (Antol et al., 2015) and visual dialog (Das et al., 2017; Mostafazadeh et al., 2017; De Vries et al., 2017).

Visual Storytelling: Huang et al. (2016) ventured into sequential step wise generation of stories by introducing visual storytelling (ViST). Chandu et al. (2019) also proposed a dataset of 16k recipes in a similar form. Recent methods have tackled ViST using adversarial learning, reinforcement learning (Wang et al., 2018; Huang et al., 2019; Hu et al., 2019), modality-fusion (Smilevski et al., 2018), traditional seq2seq models (Kim et al., 2018; Jung et al., 2020; Hsu et al., 2018) and explicit structures (Bosselut et al., 2016; Bisk et al., 2019). Though the stories in ViST demonstrate a sense of continuity, the overarching sequential context is feeble. Procedures such as cooking recipes (Salvador et al., 2019; Wang et al., 2019) on the other hand, demonstrate this characteristic inviolably. Hence, we present a large scale ViPT dataset to encourage research in this direction.

Infilling and Masking: The idea is motivated by cloze tasks (Taylor, 1953) that addresses readability and understanding. However, recent advances in learning a masked language model (Devlin et al., 2019) paved way for a new trend in exploring masked contexts (Song et al., 2019; Lewis et al., 2019). Generation of meaning patches with missing portions of text is experimented by Zhu et al. (2019); Donahue et al. (2020); Fedus et al. (2018) to generate meaningful patches.Similarly, Ippolito et al. (2019) proposed a hierarchical model to generate middle span using a bag of predicted words from left and right contexts. In a similar spirit, this paper studies the effects of infilling techniques for visual narrative generation. An alternate stream of work to improve the context in stories include providing supporting information such as entities (Clark et al., 2018; Xu et al., 2018), latent templates (Wiseman et al., 2018), knowledge graphs (Yang et al., 2019), etc., explicitly. In contrast to this, infilling provides an opportune platform to implicitly learn the contextual information. Our work is positioned in the intersection of infilling and multimodal language generation.

3 Dataset Description

This section describes our new ViPT dataset and highlights the differences with ViST.

Procedures vs Stories: Narrative properties such as content and structure in these forms are sufficiently contrastive (Gatt and Krahmer, 2018). Content in stories include characters and events while procedures include ingredients, materials and actions. While stories start from setting an era to characterizing participants, procedures start by listing ingredients followed by a step by step description.

Dataset	Stories				Recipes			
Model	XE	V-Infill	V-InfillR	INet	XE	V-Infill	V-InfillR	INet
BLEU-1	62.05	61.58	61.84	**63.31**	28.61	**29.73**	28.61	25.10
BLEU-2	38.31	37.27	37.81	**39.60**	16.89	**17.50**	17.01	13.36
BLEU-3	22.68	21.70	22.42	**23.62**	10.50	**10.83**	10.59	6.51
BLEU-4	13.74	12.96	13.69	**14.30**	5.68	**5.81**	5.71	3.60
METEOR	35.01	34.53	35.08	**35.57**	26.72	27.26	**27.51**	25.62
ROUGE_L	29.66	29.12	29.65	**30.14**	21.64	**22.02**	18.66	20.43

Table 2: Performance of different models on stories (from ViST) and recipes (from ViPT) datasets

ViPT dataset: Though stories have the potential to exhibit the properties listed above, it is challenging to observe them in ViST dataset (Huang et al., 2016) owing to the shorter sequence lengths. To facilitate multi-domain research with stronger interleaved contexts between surrounding steps, we present a large scale *visual procedure telling* dataset with 46k procedures comprising of 340k pairwise images and textual descriptions. It is carefully curated from a number of *how-to* blogging websites. Our dataset comprises of pairwise images and textual descriptions of the corresponding images, typiccally describing a step in a procedure. This means that each description of the step is tethered to an image. We categorized the dataset into 10 distinct domains including recipes, crafts, outdoors, lifestyle, technology, styling, fitness, hobbies, pets and miscellaneous. The category wise details of the dataset are presented in Table 1.

Differences between ViPT and ViST datasets: As observed in Table 1, the average number of steps in ViPT is higher than ViST. The average number of words per step in ViPT is also much higher, thereby presenting a more challenging long form text generation task. Typically, each step in the ViPT dataset comprises of multiple sentences as opposed to ViST which is a single sentence. These long sequences also present a case for dealing with larger vocabularies as well. The recipes category alone has a vocabulary of 109k tokens while the same for stories is 25k. We also compared the diversity in vocabulary of each step by computing the average percentage of unique words in a step with respect to the rest of the narrative. While this number is a high 60% for ViST, it is 39% for ViPT. This reveals the stronger cohesive and overlapping contexts in the ViPT dataset, as compared to the ViST datasets. These overlapping contexts motivates the idea of generating a sentence by bridging the contexts from surrounding sentences. Hence it forms a suitable test bed to learn interpolation with infilling technique.

4 Models Description

We present infilling based techniques for learning missing visual contexts to generate narrative text from a sequence of images. As the ViST and recipes category in ViPT are of comparable sizes (both in terms of data size and the average number of steps per instance), we perform comparative experimentation on these two categories. For our ViPT category, We use 80% for training, 10% for validation and 10% for testing. An overview of infilling based training is depicted in Figure 1. The underlying encoding and decoding stages are described here.

Encoding: Models 1, 2 and 3 here show different variants of encoding with and without infilling.

1. XE (baseline): We choose a strong performing baseline model based on sequence to sequence modeling with cross entropy (XE) loss inspired from Wang et al. (2018). The visual features are extracted from the penultimate layer of ResNet-152 by passing the resized images ($\{I_1, I_2, ..., I_n\}$) of size 224 X 224. These represent the image specific local features ($\{l_1, l_2, ..., l_n\}$). These features are then passed through a bidirectional GRU layer to attain narrative level global features ($\{g_1, g_2, ..., g_n\}$) constituting the narrative context layer in Figure 1.

2. V-Infill: We introduce an infilling indicator function on the underlying XE model by randomly sampling an infilling index (in_{idx}). This is used to construct the final infilled local features as follows.

$$l_k(\forall k, s.t. 0 < k \leq n) = \begin{cases} zero\_tensor & \text{if k=}in_{idx} \\ l_k & \text{otherwise} \end{cases}$$

Other than the sampled in_{idx}, the rest of the local features for other indices remain the same. The local features for in_{idx} are all masked to a zero tensor. The dropout of an entire set of local features from an image forces the model to learn to bridge the context from the left and the right images of in_{idx}. In this way, the infilling mechanism encourages our underlying seq2seq model to learn the local

Infill Index	**0**		**1**		**2**		**3**		**4**		**5**	
Model	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**
BLEU-1	20.9	29.7	22.8	29.8	23.5	29.9	24.4	30.4	25.5	31.0	**26.4**	**31.5**
BLEU-2	12.5	18.0	13.2	17.6	13.6	17.5	14.2	17.8	14.9	18.2	**15.4**	**18.6**
BLEU-3	7.9	11.1	8.2	10.7	8.4	10.8	8.8	10.9	9.2	11.1	**9.6**	**11.4**
BLEU-4	4.2	5.8	4.2	5.6	4.4	5.6	4.7	5.7	4.9	5.8	**5.1**	**6.0**
METEOR	27.6	27.8	26.4	27.1	26.0	26.9	26.3	27.1	26.6	27.2	26.8	27.4
ROUGE_L	20.9	22.4	20.3	21.8	20.6	21.8	21.0	21.9	21.3	22.0	**21.5**	22.1

Table 3: Performance of infilling during inference for recipes in Visual Procedure Telling

Infill Index	**0**		**1**		**2**		**3**		**4**	
Model	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**	**XE**	**V-Infill**
BLEU-1	60.9	63.0	60.8	62.0	60.3	61.9	60.5	62.2	**61.8**	**63.3**
BLEU-2	37.0	39.5	36.9	38.6	37.0	38.4	37.0	38.7	**38.1**	**39.6**
BLEU-3	21.7	23.7	21.6	23.1	21.8	22.9	21.8	23.2	**22.5**	**23.7**
BLEU-4	13.1	14.4	13.1	14.1	13.2	13.9	13.3	14.3	**13.8**	**14.5**
METEOR	34.9	35.4	34.8	35.1	35.2	35.2	35.1	35.3	**35.2**	**35.5**
ROUGE_L	29.3	30.2	29.2	29.9	29.1	30.0	29.2	30.0	**29.5**	**30.3**

Table 4: Performance of infilling during inference for Visual Story Telling

representation from contextual global features in the narrative context layer.

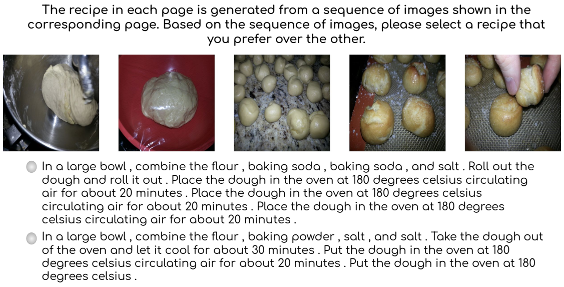
3. V-InfillR: This model varies the **R**ates in which local features are masked as training proceeds based on the indicator function above in the V-Infill model. Scheduling the number of missing features itself is a hyperparameter and we used the following setting. In the first quarter of training epochs, none are masked, then increasing it to 1 local feature for the next quarter and leaving it at 2 for the last two quarters.

Decoding: In all the above models, g_k are fed into a GRU decoder to predict each word ($\hat{\boldsymbol{w}}_t$) of the step (k). We perform beam search with a beam size of 3 during inference.Here τ is the number of words in each step and t is the current time step.

$$\hat{\boldsymbol{w}}_t \sim \prod_{\tau} Pr(\hat{\boldsymbol{w}}_t^{\tau}|\hat{\boldsymbol{w}}_t^{<\tau}, \boldsymbol{g}_k)$$

4. INet: We reimplemented the model achieving the state of the art results (Hu et al., 2019) on the visual storytelling dataset. Additionally, they use a relational embedding layer that captures relations across spatio-temporal sub-spaces. Our replication of their model is close, achieving a 35.5 METEOR and 63.3 BLEU-1 in comparison to the scores reported in their paper which are 35.6 and 64.4.

Hyperparameter Setup: We use a GRU with hidden dimension of 256 for encoder and 512 for decoder. The word embedding dimension is 512. The learning rate is 4e-4 optimized with Adam and smoothing of 1e-8. We use a dropout of 0.2 and momentum of 0.9 with a gradient clipping of 10.



The recipe in each page is generated from a sequence of images shown in the corresponding page. Based on the sequence of images, please select a recipe that you prefer over the other.

○ In a large bowl , combine the flour , baking soda , baking soda , and salt . Roll out the dough and roll it out . Place the dough in the oven at 180 degrees celsius circulating air for about 20 minutes . Place the dough in the oven at 180 degrees celsius circulating air for about 20 minutes . Place the dough in the oven at 180 degrees celsius circulating air for about 20 minutes .

○ In a large bowl , combine the flour , baking powder , salt , and salt . Take the dough out of the oven and let it cool for about 30 minutes . Put the dough in the oven at 180 degrees celsius circulating air for about 20 minutes . Put the dough in the oven at 180 degrees celsius .

Figure 2: Human Evaluation Interface for an example of generated recipes with both techniques.

The performance when experimented with a transformer based encoder along with autoregressive decoding is comparatively lesser and hence we proceed with a GRU based model. Based on the average number of steps in recipes from Table 1 which is 5.96, we truncate the recipes to 6 steps.

5 Results and Discussion

In this section, we present the effects of infilling both during both training and inference on ViST and ViPT datasets.

Infilling during training: The overall performance of the models is presented in Table 2. Both the infilling model variants achieve higher scores on the recipes while not decreasing their performances on stories. We also observed that increasing the number of masked local features beyond 2 drastically decreases the performance on both datasets.

Infilling during inference: Acquiring parallel pairwise image and narrative data in the wild is

		0	1	2	3	4	5	***Analysis***
GT		**There are only a few components necessary to sear meat and make a pan sauce.**	**Trim the fat around the meat until almost none is left .**	**Dry the meat by blotting it with paper towels .**	**Heat a little oil on a stainless steel or cast iron skillet on high heat and wait for the oil to start smoking .**	**After 3-5 minutes pass , use a spatula to check the meat.**	**Once the meat is to your desired temperature , rest on a plate and cover with foil , 5-10 minutes .**	
XE	0	Cut the chicken into small pieces .	The first thing you will need to do is put the meat in the pan .	Heat the oil in a pan and add the olive oil .	Add the salt and pepper to the pan .	Once the chicken is cooked , remove the chicken from the oven and let it cool for about 10 minutes .	Now that you have a hot pan , it 's time to cook the chicken .	- The content step 1 is being copied to step 0. - Recovering from incorrect steps until step 2.
	4	For this recipe you will need the following ingredients : - 1 cup of water , salt and pepper	Cut the chicken breast into small pieces .	The first thing you need to do is put the meat in the pan and let it cool for about 10 minutes .	Place the pan on the stove and heat the pan on medium heat .	Make sure it is properly cooked.	Once the chicken is done , remove the chicken from the oven and let it cool for a few minutes	- The infilled sentence at step 4 is generic and underspecified
V-Infill	0	You will need the following ingredients	Cut the chicken breast into small pieces .	Add the chicken to the pan .	Heat the oil in a frying pan over medium heat .	Once the chicken is done , add the chicken to the pan and stir it up .	Now that you have a little bit , you will need to put the meat in the oven at 180 degrees celsius circulating air for about 20 minutes .	+ Step 0 talks about ingredients. - The ingredients are still underspecified.
	4	You will need the following ingredients : - meat , 1 cup of butter and salt	Cut the onion into small pieces .	Cut the chicken breast into small pieces .	Heat the oil in a frying pan over medium heat .	Once the chicken is ready , add the chicken to the pan .	Once the chicken is done , remove the pan from the pan and let it sit for about 10 minutes .	+ Step 0 talks about ingredients. - Phrase from step 5 copied to step 4

Figure 3: Comparison of V-Infill and XE dealing with infilling context during inference (for making *chicken roast*). GT corresponds to the ground truth step. The index in each row corresponds to the index of the missing image.

often not feasible. Hence, we perform infilling not only at train time but also at inference time to evaluate the ability of the model to bridge contexts when the corresponding image is absent and deal with real world data imputation scenarios. Figure 3 demonstrates an example of generated samples by infilling different indices. As observed, the XE model either generates generic sentences to infill content or copies the content from the surrounding contexts. In contrast, the V-Infill model is able to generate new sentences with small phrases of sentences being copied. Underspecification of content is an issue in both these models where there is a room for improvement.

Lengths of generated sequences : We compare infilling during inference between baseline XE model and our V-Infill model in Table 3. While the METEOR scores remain comparable, the BLEU scores steadily increase as we move the in_{idx} to the right. Specifically, these jumps are bigger after step 3. Quantitatively, this is the result of the model being able to produce longer sequences as we move to the right as BLEU gets penalized for short sentences. Qualitatively, this implies that the initial steps like specifying the ingredients are more crucial as compared to later ones. A similar observation emerges by analyzing the effects of infilling during training. The average length of generated recipes by XE is 71.26 and by V-Infill is 76.49. A similar trend is observed for stories in Table 4.

Human Evaluation: We also conducted human evaluation (interface in Figure 2) in the form of preference testing comparing our V-Infill model with XE by randomly sampling 10 recipes. Among 20 evaluators, our V-Infill model is preferred 64% of the times over the XE model on an average.

6 Conclusions and Future Work

We demonstrate that infilling is a simple yet effective technique and a step towards maximizing the utilization of surrounding contexts in visual narratives. Infilling is the strategy of enabling the model to learn surrounding contextual information by masking spans of input while the decoding attempts in generating the entire text. To experimentally support our hypothesis, we collect a new large scale ViPT dataset of 46k procedures comprising 10 categories. We compare the performance of our model and conclusively show the higher significance of infilling based techniques in visual procedures compared to visual stories. In future, we plan to explore the following two directions: (1) interpolating the contexts between consecutive steps by introducing a new infilled image, and (2) addressing the underspecification problem by controlling the content in infilled image with explicit guidance. These infilling techniques are also immensely useful when dealing with data imputation with missing contexts and collaborative authoring in real world scenarios.

References

Prithviraj Ammanabrolu, William Broniec, Alex Mueller, Jeremy Paul, and Mark O Riedl. 2019. Toward automated quest generation in text-adventure games. *arXiv preprint arXiv:1909.06283*.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

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.

Yonatan Bisk, Jan Buys, Karl Pichotta, and Yejin Choi. 2019. Benchmarking hierarchical script knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4077–4085. Association for Computational Linguistics.

Antoine Bosselut, Jianfu Chen, David Warren, Hannaneh Hajishirzi, and Yejin Choi. 2016. Learning prototypical event structure from photo albums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Khyathi Chandu, Eric Nyberg, and Alan W Black. 2019. Storyboarding of recipes: Grounded contextual generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6040–6046.

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)*, volume 1, pages 2250–2260.

Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. 2017. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2.

Harm De Vries, Florian Strub, Sarath Chandar, Olivier Pietquin, Hugo Larochelle, and Aaron C Courville. 2017. Guesswhat?! visual object discovery through multi-modal dialogue. In *CVPR*, volume 1, page 3.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*.

Ruo-Ping Dong, Khyathi Raghavi Chandu, and Alan W Black. 2019. Induction and reference of entities in a visual story. *arXiv preprint arXiv:1909.09699*.

John J Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and precise touch-based text entry for head-mounted augmented reality with variable occlusion. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 25(6):30.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2650–2660. Association for Computational Linguistics.

William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. Maskgan: Better text generation via filling in the _. In *International Conference on Learning Representations*.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Spandana Gella, Mike Lewis, and Marcus Rohrbach. 2018. A dataset for telling the stories of social media videos. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 968–974.

Aleksandra Hollingshead. 2018. Designing engaging online environments: Universal design for learning principles. In *Cultivating diverse online classrooms through effective instructional design*, pages 280–298. IGI Global.

MD Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):118.

Chao-Chun Hsu, Szu-Min Chen, Ming-Hsun Hsieh, and Lun-Wei Ku. 2018. Using inter-sentence diverse beam search to reduce redundancy in visual storytelling. *arXiv preprint arXiv:1805.11867*.

Junjie Hu, Yu Cheng, Zhe Gan, Jingjing Liu, Jianfeng Gao, and Graham Neubig. 2019. What makes a good story? designing composite rewards for visual storytelling.

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, et al. 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.

Daphne Ippolito, David Grangier, Chris Callison-Burch, and Douglas Eck. 2019. Unsupervised hierarchical story infilling. In *Proceedings of the First Workshop on Narrative Understanding*, pages 37–43.

Yunjae Jung, Dahun Kim, Sanghyun Woo, Kyungsu Kim, Sungjin Kim, and In So Kweon. 2020. Hide-and-tell: Learning to bridge photo streams for visual storytelling.

Taehyeong Kim, Min-Oh Heo, Seonil Son, Kyoung-Wha Park, and Byoung-Tak Zhang. 2018. Glac net: Glocal attention cascading networks for multi-image cued story generation. *arXiv preprint arXiv:1805.10973*.

Kiyoshi Kurihara, Atsushi Imai, Nobumasa Seiyama, Toshihiro Shimizu, Shoei Sato, Ichiro Yamada, Tadashi Kumano, Reiko Tako, Taro Miyazaki, Manon Ichiki, et al. 2019. Automatic generation of audio descriptions for sports programs. *SMPTE Motion Imaging Journal*, 128(1):41–47.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin and FJ Och. 2004. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir Workshop*.

Xiaoxiao Liu, Qingyang Xu, and Ning Wang. 2019. A survey on deep neural network-based image captioning. *The Visual Computer*, 35(3):445–470.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. *arXiv preprint arXiv:1701.08251*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Amaia Salvador, Michal Drozdzal, Xavier Giró-i-Nieto, and Adriana Romero. 2019. Inverse cooking: Recipe generation from food images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10453–10462. Computer Vision Foundation / IEEE.

Marko Smilevski, Ilija Lalkovski, and Gjorgi Madzarov. 2018. Stories for images-in-sequence by using visual and narrative components. *arXiv preprint arXiv:1805.05622*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

Hao Wang, Doyen Sahoo, Chenghao Liu, Ee-Peng Lim, and Steven C. H. Hoi. 2019. Learning cross-modal embeddings with adversarial networks for cooking recipes and food images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11572–11581. Computer Vision Foundation / IEEE.

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.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187.

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315.

Pengcheng Yang, Fuli Luo, Peng Chen, Lei Li, Zhiyi Yin, Xiaodong He, and Xu Sun. 2019. Knowledgeable storyteller: a commonsense-driven generative model for visual storytelling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5356–5362. AAAI Press.

Wanrong Zhu, Zhiting Hu, and Eric Xing. 2019. Text infilling. *arXiv preprint arXiv:1901.00158*.

7 Supplemental Material

1. Human Evaluation: Figure 2 depicts a screenshot of our human evaluation interface. A sequence of images are presented on top of the screen. This evaluation is conducted to compare between XE and V-Infill model. The generated sentences from the 2 models, in this case XE and V-Infill are presented after the images. Note that the generated outputs are presented in arbitrarily random order for each example to ensure there is no bias while performing preference testing. Human subjects are asked to pick one of the generated recipes for the given sequence of images. 10 such recipes are presented for each user and we avergaed the preference scores among 20 users.

2. Data Collection Process: We manually examined around 10 blogging websites with various user written text on several *how-to* activities. Among these we found that *snapguide* and *instructables* are consistent in the form of pairs of textual descriptions along with their images. We are going to release the scripts used to collect this data as well as preprocess them. We removed all the procedures in which atleast one image in each step is absent. Once all this preprocessing is done, the data contained the following categories in both the websites. These categories are based on the tags given by the bloggers to the articles they have written from among the categories that each website offers. These categories for each of these websites are:

- *snapguide:* recipes, games-tricks, sports-fitness, gardening, style, lifestyle, outdoors, beauty, arts-crafts, home, music, photography, pets, automotive, technology

- *instructables:* crafts, cooking, teachers, circuits, living, workshop, outside

In union, they are a total of 18 categories. We manually examined a few procedures in each of the categories and regrouped them into 10 broad categories that are presented in Table 1. A list of urls corresponding to the data is submitted along with the paper.

3. Visualization of topics: Each of the categories in our Visual Procedure Telling (ViPT) are analyzed for the topics present in them. A screenshot highlighting one of the topics in recipes category is presented in Figure 4. To get a more detailed understanding of these topics in the dataset, we hosted the topic visualizations here: visual-narratives.github.io/Visual-Narratives/.

category	snapguide	instructables
recipes	desserts, food	cooking
crafts	arts-crafts	craft
outdoors	outdoors, gardening	outside
lifestyle	lifestyle, home	living
technology	technology, automotive	circuits
styling	style, beauty	
fitness	sports-fitness	
hobbies	music, photography	
pets	pets	
misc	games-tricks	teachers, workshop

Table 5: Regrouping the categories in ViPT dataset

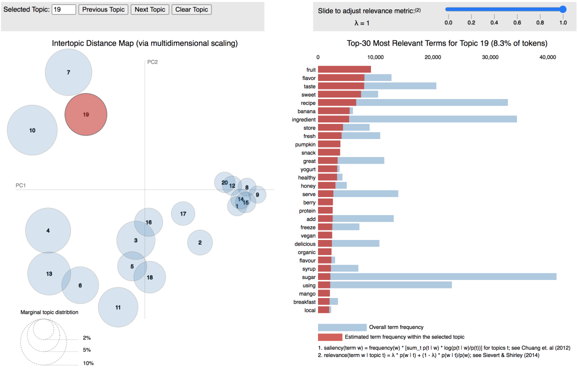

Selected Topic: 19
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
7
19
10
20
12
8
9
14
15
1
17
16
4
3
2
5
13
6
18
11
Marginal topic distribtion
2%
5%
10%
Top-30 Most Relevant Terms for Topic 19 (8.3% of tokens)
0
10,000
20,000
30,000
40,000
fruit
flavor
taste
sweet
recipe
banana
ingredient
store
fresh
pumpkin
snack
great
yogurt
healthy
honey
serve
berry
protein
add
freeze
vegan
delicious
organic
flavour
syrup
sugar
using
mango
breakfast
local
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 4: Visualization of topic model for recipes category

4. Performance of infilling during inference for Visual Story Telling: Table 4 demonstrates the effects of infilling various indices during inference. This table is analogous to Table 3 for stories. As we can see, a similar trend in the increase in all the automatic metrics are present as we move the infill index to the right of the story. While that is still the case, a very interesting observation is that the difference between the performance of XE and Infill models for any given index is much higher for recipes compared to stories. The infilling technique is bringing much more value to the task when the nature of the text is procedural and dependent more on the surrounding contexts.

5. Reproducibility Checklist:

1. **For all reported experimental results:**

   (a) *A clear description of the mathematical setting, algorithm, and/or model:* Yes. There are two important definitions to understand in the paper. The first is the task definition which is outlined in Section 1. The second is the infilling technique definition which is presented in Section

Hyperparameter	Value
Num of Epochs	100
Batch size	64
Learning rate	4e-4
Momentum	0.9
Gradient clipping	10
Dropout	0.2
Optimizer	Adam
alpha for Adam	0.8
beta for Adam	0.99
Adam smoothing	1e-8
Word Embedding dim	512
Visual feature dim (ResNet)	2048
Num of GRU layers	1
GRU Hidden dim	512
Beam width	3

Table 6: Hyperparameters of the best performing model

4.

(b) *Submission of a zip file containing source code, with specification of all dependencies, including external libraries, or a link to such resources (while still anonymized):* No. Though they are not submitted along with the paper, we plan to release both the code and the data along with the scripts to preprocess the data here: visual-narratives.github.io/Visual-Narratives/

(c) *Description of computing infrastructure used:* The models are trained on a GPU with driver version 430.26 and CUDA version 10.2. The GPU RAM being used have a maximum size of 8GB and 12 GB for different experiments. A few of the experiments are run on GeForce GTX 1070 with 1920 cores and GeForce GTX 1080 Ti with 3584 cores.

(d) *Average runtime for each approach:* On an average the experiments on the visual procedure telling dataset took around 8 hours and the same on the visual story telling dataset took around 11 hours. These numbers are the averages of the 4 different models on each dataset comprising of XE, V-Infill, V-InfillR and INet.

(e) *Number of parameters in each model:* For the models pertaining to XE, V-Infill and V-InfillR, the number of parameters are listed below. For the ViPT dataset, the number of trainable parameters are 15,665,896. For the ViST dataset, the number of trainable parameters are 14,390,894. The INet model for the ViPT dataset has 19,345,128 and on the stories data is 18,070,126. As we can see, the infilling variants are able to achieve better performance on procedural texts with comparatively fewer parameters. This justifies our hypothesis that infilling is a simple yet effective technique to perform text generation where there are strong contextual dependencies.

(f) *Explanation of evaluation metrics used, with links to code:* The automatic metrics are evaluated using BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005) and ROUGE_L (Lin and Och, 2004). The scripts used to evaluate will be released along with the code. We adapted the standard code that computes the aforementioned metrics for the task of MS COCO image captioning challenging for our task. The code for the metrics can be found here: github.com/microsoft/DialoGPT/tree/master/pycocoevalcap. The code in the above repository is not ours. We simply used their evaluation scripts for our task.

2. **For all experiments with hyperparameter search:**

(a) *Bounds for each hyperparameter:* For the scheduling of the infilling, we tried the following settings. We experimented with the following settings. The first setting is: (i) 0 images for first quarter (ii) 1 image for second quarter (iii) 2 images for the last half number of epochs. The second setting is: (i) 0 images for first quarter (ii) 1 image for 2 quarters (iii) 2 images for the last quarter number of epochs. The third setting is: (i) 1 images for first quarter (ii) 2 image for second quarter (iii) 3 images for the last half number of epochs. The fourth setting is: (i) 0 images for first quarter (ii) 1 image for second quarter (iii) 2 images for the third quarter (iv) 3 images for the last quarter number of epochs. For the rest of the hyperparameters, we used the same as the baseline. We contacted the authors of the INet model, which is the state-of-the-art on visual storytelling for the hyperparameter values they have used and used the same in our work to replicate their results and perform reliable comparisons.

(b) *Hyperparameter configurations for best-performing models:* The hyperparameters for the best performing model are descibed in Section 4. A more detailed set of the hyperparameters are presented in Table 6.

(c) *Number of hyperparameter search trials:* The important hyperparameter that we tuned is the scheduling of infilling. The hyperparam that performed best is described in Section 4. In total, we have tried 4 settings to arrive pick the best setting.

(d) *The method of choosing hyperparameter values (e.g., uniform sampling, manual tuning, etc.) and the criterion used to select among them (e.g., accuracy):* We performed manual tuning in various settings for the rate of infilling in the V-InfillR model. The criterion used to select the best model is METEOR score, which is the automatic metric to evaluate our model.

(e) *Expected validation performance, or the mean and variance as a function of the number of hyperparameter trials:*

3. **For all datasets used:**

(a) *Relevant statistics such as number of examples:* The statistics of the ViPT dataset along with category wise distribution for the number of procedures, number of steps, average number of steps and average number of words per step are presented in Table 1. The same table also compares these distributions with the ViST dataset.

(b) *Details of train/validation/test splits:* The splits to perform training, validation and testing are readily available for ViST dataset. For our newly collected ViPT dataset, we conduct experiments on the recipes category. These splits are done for this dataset as 80% for training, 10% for validation and 10% for testing.

(c) *Explanation of any data that were excluded, and all pre-processing steps:* The data is cleaned of any html tags as they are collected from the blogging websites.

(d) *A link to a downloadable version of the data:* We plan to release the preprocessed data along with visual features. It will be avaialable for download here: visual-narratives.github.io/Visual-Narratives/

(e) *For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control:* We collected the data from *how-to* blogging websites. From the collected procedures, we perform an additional step of removing the ones that do not have any images. We keep the procedures in the final dataset that have at least one image per step. Additionally, we manually inspected the different types of procedures and categorized them into the 10 categories as mentioned in the paper.