Text2Scene: Generating Compositional Scenes from Textual Descriptions

Parm Johal
parmj@uvic.cag

Sean McFarlane
seanmcf@uvic.ca

Simon Walker
walker11@uvic.ca

University of Victoria
3800 Finnerty Road, Victoria, BC

Abstract

In this paper, we extend the proposal of Text2Scene, a model generating different types of compositional scene representations using natural language descriptions. In this extended experimentation, we start off by using long short-term memory (LSTM) instead of gated recurrent units (GRU) for image encoding. We focus our experimentation on an altered version of the abstract scenes dataset to exhibit the similarities and differences in accuracy and computation between the two networks. We then proceed to combine the LSTM recurrent units with a different network architecture, called ResNeXt, in place of the ResNet architecture used in the original experiments. After combining these 3 alterations, we are able to obtain competitively accurate results when compared to the original model.

1. Introduction

Our project takes the original Text2Scene model and enhances some parts that have been outdated due to recent discoveries. After receiving the original code, running it took hours to compute a few epochs. We decided that the first improvement to make would be to speed up the execution of the network. Motivated by this, we decided to use a new architecture called ResNeXt which would increase the speed at which each epoch processes the data (figure 6). We further worked on implementing our own data set by replacing objects in the scene (table 1) and setting up TensorBoard to allow us to monitor accuracy and loss. Due to problems on the user side we were not able to fully implement the TensorBoard.

Images are encoded in the original Text2Scene by using gated recurrent units (GRU) and long short-term memory (LSTM). GRU is a gating mechanism in recurrent neural networks used to solve the vanishing gradient problem, which is when the gradient eventually becomes insignificant and prevents the weight from changing its value. The LSTM is a better version of a GRU since it can be used with bigger datasets because of its modified gates based off of the ones in the GRU. From this, we hypothesize that the dataset we are using will work better if we encode the data using the LSTM rather than the GRU considering the amount of data that is being processed.

With these changes we are able to describe and show the improvements we contribute to the Text2Scene model. In the following sections we will discuss some related work, the original and updated model, our experiments and improvements, and finally our conclusion. Our contributions are summarized as follows:

- We show that by changing to the ResNeXt architecture we can increase the accuracy and runtime of the model.
- We generate a modified dataset to conduct experiments on.

2. Related Work

In the original project made by Tan et al.[9], they noted other related works on this topic such as generated captions from images [1, 5] and text-to-image synthesis [7, 8]. What they noticed was that even though these works gave quality information from their experiments, they were not taking into account scenes in which multiple objects interact with each other. This was why they proposed their Text2Scene model which "produces a scene by sequentially generating objects (e.g. in the forms of slip-arts, bounding boxes, or segmented object patches) containing the semantic elements that compose the scene" [9].

Some works that were closely related to the original Text2Scene are [4, 3, 2]. Johnson et al [4] used structured scene graphs to generate their images by having the objects and their details be the inputs to the graph. In Text2Scene the objects effect on the overall image (placement, orientation, etc.) is inferred from text. Hong et al [3] generated layouts as representations in a trained module unlike the work from before. Due to this their results were not as accurate as Text2Scene since they generated pixel-level outputs which gave a stronger result which is seen in the overall image. Gupta et al [2] generated cartoon-like images by having the

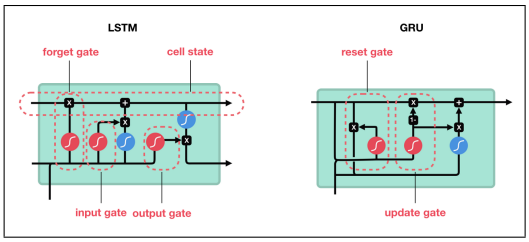

Figure 1. Gate structure comparison between LSTM and GRU.[6]

presented objects be the input to the model, and the predictions of how the scene will be generated were done by separably trained modules. Text2Scene instead does more realistic scenes and uses an "end-to-end"[9] way of training. To Tan et al [9], the Text2Scene model was one of the first to go beyond cartoon-like imagery and be capable of generating detailed scenes under a single working framework.

3. Model

3.1. Dataset

The original dataset was constructed by using turkers to generate 10,020 variations of an original 1002 manually constructed seed scenes, with 10 variations on each original scene. The result of the turking was 10,020 scene layouts stored as object coordinates in Scenes_10020.txt. The data in Scenes_10020.txt was then fed into SceneRenderer.exe, which composes each final scene image out of individual source object images based on the coordinates given. Additional metadata is constructed that is used by the network to form associations. An object list is created, which identifies certain words as entities in the scene. A relation list enumerates different ways scene objects are paired together. A bag-of-words representation for each description is created along with a list of all words sorted by their frequency. The resultant 10,020 scene images and 10,020 scene descriptions are fed into Text2Scene as input, along with the metadata, and used to train the model.

3.2. Image Encoding

In this model we use LSTM to resolve short term memory in the RNN. LSTM is used instead of GRU since it uses more training parameters and it is more accurate with datasets involving longer sequences of data [6]. This is shown in figure 1, as LSTM is shown having more gates. This leads to better control of the output and hence, better accuracy. Since the original model learns to sequentially generate objects and their attributes at each time step [9], we use LSTM to aid this task.

3.3. Pre Trained Network Architecture

The pretrained network consists of a highly modularized version of the currently established ResNet architecture.

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128, C{=}32 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3	28×28	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256, C{=}32 \\ 1\times1, 512 \end{bmatrix}\times4$
conv4	14×14	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512, C{=}32 \\ 1\times1, 1024 \end{bmatrix}\times6$
conv5	7×7	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024, C{=}32 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**×10^6	**25.0**×10^6
FLOPs		**4.1**×10^9	**4.2**×10^9

Figure 2. Resnet vs. ResneXt [9]

The network is developed by repeating a building block that sums a set of transformations with the same topology [10], which is shown in figure 2. Unlike the original ResNet architecture, the ResNeXt architecture keeps track of the number of paths it goes through, using 'C' as the variable above.

4. Experiments

Since time constraints were a significant issue, tests were limited to between 1 and 5 epochs. Tests were performed on the original dataset and model, the altered dataset with LSTM, the original dataset with ResNeXt and LSTM, and the altered dataset with ResNeXt and LSTM.

4.1. Changing Architecture

After running the original model, we proceed to run the same model with the LSTM + ResNeXt architectural change. The results can be seen in figure 3. The move to a new architecture significantly improved the efficiency of the training, but at the cost of accuracy. The training accuracy dropped by 0.6% and the validation accuracy by 1.1%. However, we are able to see an increase in accuracy for the coordinate validation field by 0.17%.

4.2. Adding Altered Dataset

Due to feasibility constraints an entirely new dataset could not be constructed. The original dataset makes extensive use of turking in its creation which puts it outside the scope of what we have available. As an alternative, the original scene layouts were used, but with certain objects replaced in every scene they appear. Additionally, any textual references to these objects are replaced. This would, at the very least, allow us to test the neural network on novel

Epochs = 5 (times)	**Original Run (711m28s)**		**LSTM + ResNeXt (248m26s)**		**LSTM's + ResNeXt + Altered Dataset (248m20s)**	
Fields	Avg acc	Avg err	Avg acc	Avg err	Avg acc	Avg err
Train	0.478	4.9	0.472	5.28	0.467	5.39
Valid	0.473	4.98	0.462	5.25	0.456	5.42
Obj Valid	0.573	-	0.526	-	0.473	-
Pose Valid	0.395	-	0.378	-	0.386	-
Expr Valid	0.427	-	0.398	-	0.412	-
Coord Valid	0.0446	-	**0.0463**	-	0.0451	-
Scale Valid	0.775	-	0.771	-	0.763	-
Flip Valid	0.644	-	0.65	-	**0.655**	-

Figure 3. Results after training the revised model.

Original Object	Replacement Object
Baseball bat	Katana
Pirate hat	Top hat
Pie	Cake
Snake	Armadillo
Dog	Duck

Table 1. Objects that were replaced in the modified dataset.

Replaced object	Original reference	Replaced reference
Pirate hat	...wants to be **a pirate** .	...wants to be **dapper** .
Baseball bat	...is playing **baseball** in the...	...is playing **with swords** in the...

Table 2. Phrases associated with the objects which were modified.

image and text data, and examine how it adapts with training. The modified scene objects are listed in table 1 .

The original images were replaced with new PNG files, and text references to these objects were replaced in all the textual input data. The replacements were selected to still fit the contexts the originals were used in, since the layouts of the scenes must remain the same. A script was used to automatically replace all occurrences of these objects in the text with their new versions. Additionally, some other phrases in the 10,020 scene descriptions were replaced to match the new objects, some examples of which can be seen in table 2.

After the original PNG files were replaced, the

Epochs = 5 (time)	**Original Run (711m28s)**		**LSTM + Altered Dataset (No network architecture change) (432m33s)**	
Fields	Avg acc	Avg err	Avg acc	Avg err
Train	0.478	4.9	**0.482**	4.99
Valid	0.473	4.98	**0.477**	5.06
Obj Valid	0.573	-	0.543	-
Pose Valid	0.395	-	0.392	-
Expr Valid	0.427	-	0.427	-
Coord Valid	0.0446	-	**0.0510**	-
Scale Valid	0.775	-	**0.778**	-
Flip Valid	0.644	-	**0.669**	-

Figure 4. Results from using only LSTM with the altered dataset.

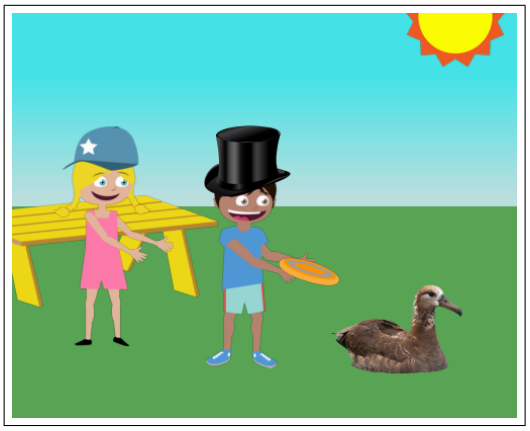

Figure 5. A generated scene using the new "Top Hat" and "Duck" objects.

SceneRenderer executable included with the data was used, which generated 10,020 unique scenes composed from the new source images. An additional script was created to update the metadata such as the bag-of-words representation and occurrence-count word list, as no functionality to generate this data was included with the original dataset.

4.3. Results

Looking at the data between figures 3 and 4 we can see the results when the altered dataset is added. From figure 3 we see a further decrease in training and validation accuracy (with the exception of the flip validation accuracy), but when the ResNeXt network architecture is replaced with the original ResNet network and used with the altered dataset we see a big improvement. More importantly, we see that the training and validation accuracies both improve

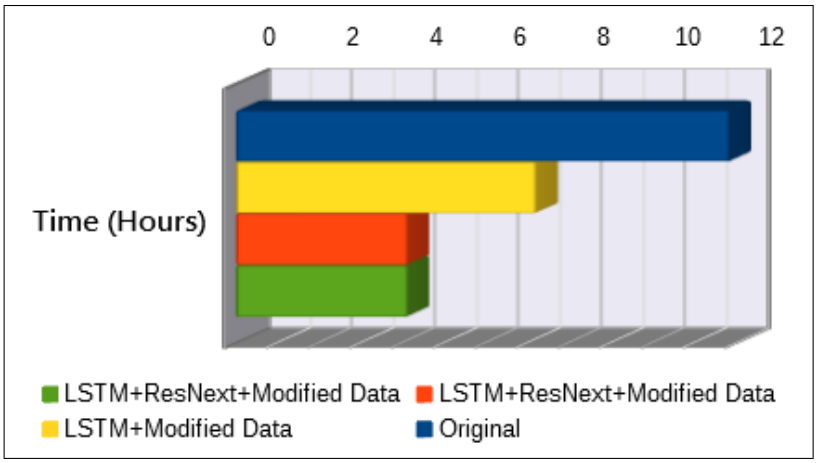

Figure 6. Total runtime over 5 epochs for each of the four tests.

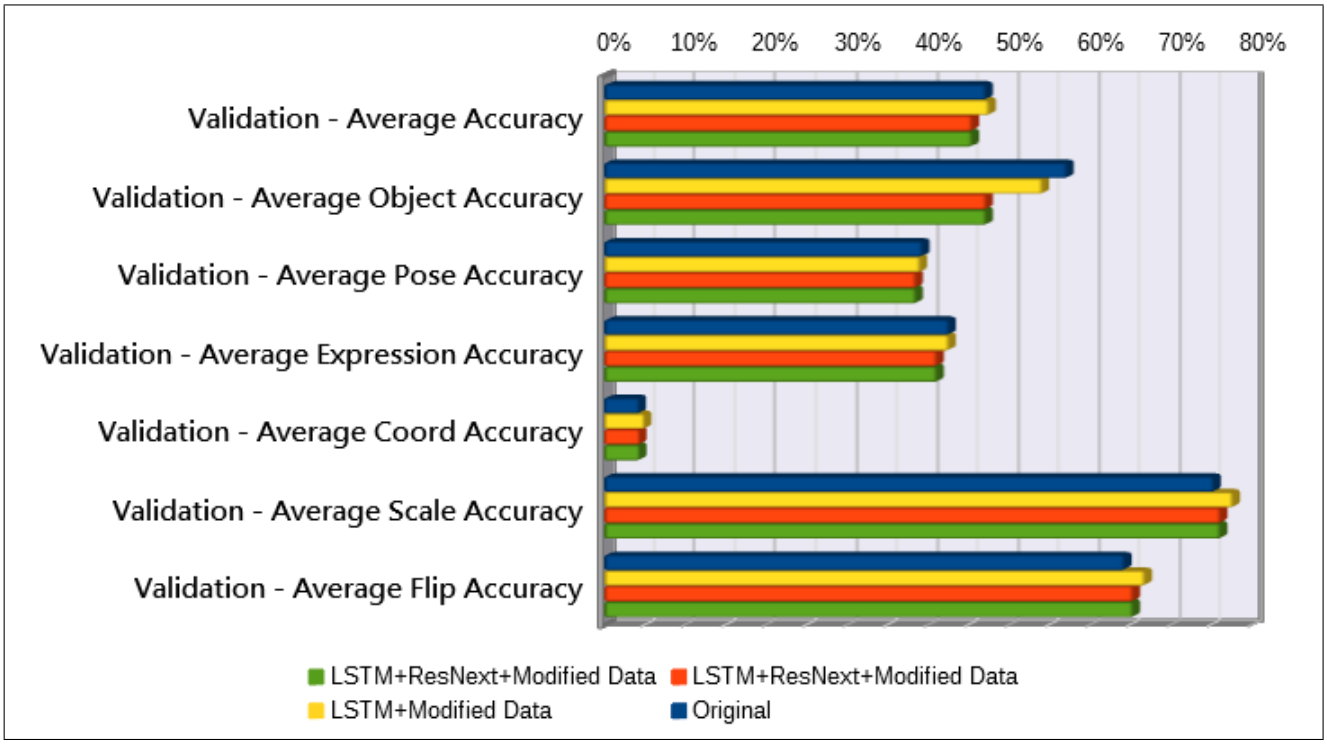

Figure 7. Validation step scores for each of the tests conducted.

by 0.4%. This was an unexpected result as we anticipated an improvement on the dataset with ResNeXt, but instead we discovered better scores using ResNet.

5. Conclusion

Our changes to the original design of Text2Scene have increased the speed and accuracy of the training while retaining the specifics of the original work. Although we initially hypothesized the switch to ResNeXt giving us an increase in accuracy for our altered dataset, the ResNet implementation combined with LSTM surprised us with the increase in accuracy. The ResNeXt architecture alongside the change of the dataset, however, made it so that we can run the model in better time due to feasibility constraints. This improved runtime may even result in increased accuracy if given the same amount of time to train.

References

[1] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. in european conference on computer vision (eccv), 2010.

[2] T. Gupta, D. Schwenk, A. Farhadi, D. Hoiem, and A. Kembhavi. Imagine this! scripts to compositions to videos. in european conference on computer vision (eccv), 2018.

[3] S. Hong, D. Yang, J. Choi, and H. Lee. Inferring semantic layout for hierarchical textto-image synthesis. in ieee conference on computer vision and pattern recognition (cvpr), 2018.

[4] J. Johnson, A. Gupta, and L. Fei-Fei. Image generation from scene graphs. in ieee conference on computer vision and pattern recognition (cvpr), 2018.

[5] R. Mason and E. Charniak. Nonparametric method for data-driven image captioning. in annual meeting of the association for computational linguistics (acl), 2014.

[6] M. Nguyen. Illustrated guide to lstm's and gru's: A step by step explanation, 2018.

[7] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. in advances in neural information processing systems (neurips), 2016.

[8] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. in international conference on learning representations (iclr), 2016.

[9] F. Tan, S. Feng, and V. Ordonez. Text2scene: Generating compositional scenes from textual descriptions, 2018.

[10] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks, 2016.